IFA-EO: An improved firefly algorithm hybridized with extremal optimization for continuous unconstrained optimization problems

Min-Rong Chen • Liu-Qing Yang • Guo-Qiang Zeng • Kang-Di Lu • Yi-Yuan Huang

Abstract

As one of the evolutionary algorithms, firefly algorithm (FA) has been widely used to solve various complex optimization problems. However, FA has significant drawbacks in slow convergence rate and is easily trapped into local optimum. To tackle these defects, this paper proposes an improved FA combined with extremal optimization (EO), named IFA-EO, where three strategies are incorporated. First, to balance the tradeoff between exploration ability and exploitation ability, we adopt a new attraction model for FA operation, which combines the full attraction model and the single attraction model through the probability choice strategy. In the single attraction model, small probability accepts the worse solution to improve the diversity of the offspring. Second, the adaptive step size is proposed based on the number of iterations to dynamically adjust the attention to the exploration model or exploitation model. Third, we combine an EO algorithm with powerful ability in local-search into FA. Experiments are tested on two group popular benchmarks including complex unimodal and multimodal functions. Our experimental results demonstrate that the proposed IFA-EO algorithm can deal with various complex optimization problems and has similar or better performance than the other eight FA variants, three EO-based algorithms, and one advanced differential evolution variant in terms of accuracy and statistical results.

Keywords Firefly algorithm • extremal optimization • Probability choice strategy • Adaptive step size • Continuous optimization problems

1 Introduction

Most real-life engineering problems are the optimization problems in essence, such as production scheduling, investment portfolio, and vehicle routing problems, which can save resources and improve work efficiency. Recently, optimization methods for solving various optimization problems have attracted much attention in academia and engineering domain. As one of the main optimization branches, the concept of swarm intelligence was first proposed in Beni and Wang (1993). Later, biologists and bionic algorithm experts proposed a swarm intelligence optimization algorithm based on swarm intelligence’s concept. It is a bio-inspired group intelligent optimization algorithm whose idea is to draw on and simulate the functions, features, phenomena, and behaviors of natural organisms (Bonabeau et al. 1999). At present, the main swarm intelligence optimization algorithms include particle swarm optimization (PSO) (Kennedy 2010), genetic algorithm (GA) (Deb et al. 2002), cuckoo search algorithm...
As a novel optimization framework, extremal optimization (EO) (Boettcher and Percus 1999, 2000) is inspired by the far-from-equilibrium dynamics of self-organized criticality (SOC). EO is based on a process where the bad species is always forced to mutate according to a uniform random or power-law probability distribution. Due to its characteristics, EO has a strong ability in local search and eliminates bad components. Therefore, EO and its variations have been successfully used in a wide variety of optimization problems such as optimal controller design (Zeng et al. 2019, 2015; Lu et al. 2019). It is worth mentioning that EO as an ancillary technique with the exploitation ability has largely improved the performance PSO (Chen et al. 2010) and ABC (Chen et al. 2019). A natural idea is to introduce EO in FA and test whether the performance of the improved algorithm can be enhanced. Inspired by the aforementioned descriptions, this paper proposes an IFA-EO algorithm, which hybridizes the advantages of FA and EO to reduce the computation time, improve convergence speed and prevent falling into local optimum.

Note that in Chen et al. (2010), the authors combined PSO with EO and proposed a novel hybrid algorithm called PSO-EO. The weakness of PSO in premature convergence is compensated by introducing the EO with strong local search capability. Compared with PSO-EO, the proposed IFA-EO firstly improves the exploration ability by using single attraction model through the probability choice strategy and then combines IFA with EO. Also, in Chen et al. (2019), the authors proposed ABC-EO, which makes full use of exploration capability of ABC and the exploitation capability of EO. Compared with ABC-EO, the proposed IFA-EO considers an adaptive step size based on the number of iterations to dynamically adjust the attention to the exploration model or exploitation model.

The main new contributions can be summarized below:

1. It is a vital role in an optimization algorithm to keep the balance between exploration ability and exploitation ability. The traditional FA uses a full attraction model focusing on global search, which causes the low convergence speed. To balance between exploitation ability and exploration ability, we adopt a novel model that combines the full attraction model and the single attraction model through a probability choice strategy. In the single attraction model, small probability accepts the worse solution to improve the search capability. Compared with PSO-EO, the proposed IFA-EO considers an adaptive step size based on the number of iterations to dynamically adjust the diversity of the offspring.

2. The standard FA adopts a fixed step size, that is, each iteration step is equal. Different from this operation, the adaptive step size is proposed based on the number of iterations. As the number of iterations increases, the step size decreases gradually. When the time is big, the step size is close to zero.

3. EO algorithm has only one mutation operation, which is simple and easy to implement. In this paper, we introduce EO algorithm to FA, in order to better prevent FA algorithm from trapping into the local optimum.
To illustrate the performance of the IFA-EO in this paper, we use two sets of unimodal/multimodal benchmark functions to assess the performance of IFA-EO by comparing with thirteen successful swarm intelligence algorithms. We also discuss the impact of different strategies and different populations on IFA-EO. The thirteen competitors include eight variants of FA, three EO-based algorithms, and one advanced DE variant. To be more specific, the eight variants of FA are new and efficient FA (NEFA) (Pan et al. 2018), FA with adaptive control parameters (ApFA) (Wang et al. 2017a), FA with neighborhood attraction (NaFA) (Wang et al. 2017b), VSSFA (Yu et al. 2015), WSSFA (Yu et al. 2014), CFA (Gandomi et al. 2013), MFA (Jr et al. 2014), standard FA (FA) (Yang 2010b) and an adaptive logarithmic spiral-Levy FA (AD-IFA) (Wu et al. 2020). Three EO-based algorithms are population-based EO (PEO) (Chen et al. 2006), PSO hybridized with EO (PSO-EO) (Chen et al. 2010), real-coded PEO algorithm with polynomial mutation (RPEO-PLM) (Li et al. 2016). Besides, one advanced competitor is linear population size reduction technique of success history-based adaptive differential evolution (L-SHADE) (Biswas et al. 2019). From the experimental results, we can see that IFA-EO has similar or better performance than the other thirteen optimization algorithms in terms of accuracy and statistical results.

This paper is arranged as follows. We briefly introduce the standard FA in Sect. 2. Section 3 describes the proposed IFA-EO algorithm with three strategies in detail. In Sect. 4, the proposed IFA-EO algorithm is employed to handle two groups of unconstrained continuous benchmark functions by comparing with other competitors. Furthermore, we investigate the effects of different strategies and population size on the performance of IFA-EO. Finally, Sect. 5 concludes this paper and gives the future work (Tang and Tseng 2013; Ho et al. 2004; Storn and Price 1997).

2 Firefly algorithm

FA is a meta-heuristic algorithm derived from swarm intelligence. The basic idea of FA is to simulate the firefly flashing behavior in nature, search for a brighter one around. It gradually moves toward the better position and gathers to the brightest position, i.e., achieving the best solution. FA has a simple concept, clear process, few parameter settings, no mutation, crossover, and other complex operations, so it is easier to operate. Here, we assume that (Yang 2010b):

1. In the algorithm, the gender difference of firefly individuals is not considered, that is, all firefly individuals are considered to be of the same sex.
2. The attraction of fireflies is based on their light intensity. Consider two fireflies, the brighter firefly will attract the weaker firefly, and the attraction will reduce as the distance between them increases.
3. In practical application, the light intensity of firefly individuals is generally associated with objective function value, and usually objective function value is considered to be the light intensity of the firefly at that point.

Algorithm 1 gives the pseudo-code of FA (Yang 2010b).

| Algorithm 1: FA |
|----------------|
| Objective function $f(x), x=(x_1, x_2, ..., x_d)^T$; |
| Initialize population $x_i\ (i=1, 2, ..., n)$ (the number of fireflies is $n$); |
| Light intensity $I_i$ at $x_i$ is depended on $f(x_i)$; |
| Determine the coefficient of light absorption $γ$; |
| While ($t < MaxIter$) // MaxIter means the maximum number of iterations; |
| For $i=1 : n$ // $n$ means the population size |
| For $j=1 : n$ |
| If ($I_i < I_j$) |
| Move firefly $i$ toward $j$; |
| End if |
| Vary attractiveness with distance $r$ via exp$(-γr)$. Evaluate new solution and then update light intensity; |
| End For |
| End For |
| Rank all the fireflies and search the current best solution $I_0$ found so far; |
| End While |
The light intensity of firefly usually reduces with the increase of distance. The light intensity $I$ is given below (Yang 2010b):

$$I(r) = I_0 e^{-\gamma r^2}$$

where $I_0$ denotes the original light intensity of the firefly, $r$ means the distance between two fireflies. $\gamma$ is called light absorption factor.

The attractiveness $\beta$ is defined as follows (Yang 2010b):

$$\beta = \beta_0 e^{-\gamma r^2}$$

where $\beta_0$ means the attractiveness when $r = 0$. The distance between fireflies $i$ and $j$ at the spatial coordinates, i.e., $x_i$ and $x_j$, can be expressed as the Cartesian distance below (Yang 2010b):

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{D} (x_{i,k} - x_{j,k})^2}$$

where $D$ denotes the number of dimension. $x_{i,k}$ and $x_{j,k}$ represent the $k$-th components of $x_i$ and $x_j$, respectively. In the 2-dimensional case, the distance can be described as follows (Yang 2010b):

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The movement of a firefly $i$ attracting to another more attractive firefly $j$ is defined as the following equation (Yang 2010b):

$$x_i^{(t+1)} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \varepsilon_i$$

where $\alpha$ is step size, and $\varepsilon_i$ means a vector of random numbers from a Gaussian distribution or uniform distribution. $\alpha$ ranges from 0 to 1, $\varepsilon_i$ is usually replaced by $(\text{rand} - 0.5)$ where rand is a random number uniformly distributed in $[0,1]$.

3 The proposed approach

The traditional FA has a slow convergence speed, premature convergence, and easily falling in the local optimum, which leads to low solution accuracy (Olamaei et al. 2013; Bhushan and Pillai 2013; Trunfio 2014). To deal with these problems, this study presents an improved FA called IFA-EO. In IFA-EO, three improvement strategies are adopted. Firstly, a probability choice strategy and small probability acceptance of the worse solution are used to improve the diversity of offspring. Secondly, adaptive step size is adopted to keep the balance between exploration ability and exploitation ability. Thirdly, extremal optimization (EO) (Chen et al. 2010) is suggested to enhance the local search capability of FA.

3.1 An attraction with probability choice strategy

Owing to the merits such as simple concept, clear process, no complexity operations, FA has attracted much attention from many scholars. Thus, many modified versions of FA have been presented in recent decades. These versions of FA are essential to strengthen the search ability and improve the accuracy of FA. On the one hand, if we pay too much attention to exploration, the convergence speed of the algorithm will slow at the end of the searching phase; on the other hand, if too much attention is paid to the exploitation process, the algorithm will easily prematurely converge. Therefore, how to balance exploration ability and exploitation ability is very important. In general, at the beginning of optimization, exploration operation is more important because we need quickly and roughly search the entire search space to find the most potential areas. To tackle this problem, we propose an attraction model with a probability choice strategy. The attraction model is described as follows.

The probability is calculated by using Eq. (6) (Lv et al. 2018):

$$P = P_{\min} + (P_{\max} - P_{\min}) \times e^{1 - \frac{MaxIter - t}{MaxIter - MaxIter - t}}$$

where $P_{\min}$ is the minimum value of attraction probability, $P_{\max}$ means the maximum value of attraction probability, $MaxIter$ means the maximum iteration number, and $t$ denotes the current iteration number. $P_{\max}$, $P_{\min}$ range from 0 to 1, $P_{\max}$, $P_{\min}$ usually set as 0.9, 0.05, respectively. As the number of iterations increases, $P$ will gradually decrease from $P_{\max}$ to $P_{\min}$.

In each iteration, a random number $q$ is obtained between 0 and 1. If $q < P$, we adopt a full attraction model (Yang 2010b), as shown in Fig. 1. We can see that the $i$th firefly is attracted to other fireflies. It is beneficial to perform exploration at an early stage. Otherwise, if $q > P$, single attraction model is used, as shown in Fig. 2. In the single attraction model, the $i$th firefly is attracted to only

![Fig. 1 Full attraction model (Yang 2010b)](image)
one neighbor $i$-1th firefly, which is a better solution than the $i$th firefly. Emphasize that $x_{i-1}$ in Fig. 2 is $i$-1th firefly sorted according to the descending order of light intensity. It makes the convergence speed very fast at the later period.

In the single model, it may only focus on local search, leading to premature convergence. In order to jump out of local optimum, inspired by the basic idea of simulated annealing, we accept some worse solutions with a small probability in the single model, that is, fireflies have chances to move towards darker fireflies to avoid falling into local optimum.

### 3.2 Adaptive step size

In the standard FA, using a fixed step size is not conducive to balancing exploration and exploitation. When the step size is set too large, the search at the later stage may skip the optimal solutions, otherwise, the convergence speed is too slow. The steps size of FA should be tuned according to actual conditions. To solve this problem, Yu et al. (Yu et al. 2015) proposed VSSFA and WSSFA, respectively. Furthermore, in Wang et al. 2017a, Wang et al. discussed the relationship between the rate of convergence and step size $\alpha$. If FA is convergent, the parameter $\alpha$ will satisfy the condition as follows (Wang et al. 2017a):

$$\lim_{t \to \infty} \alpha = 0$$

(7)

Inspired by Eq. (7), we propose an adaptive step size $\alpha$ by using a nonlinear equation. The ability to balance exploration and exploitation will be beneficial, and it should also focus on its current iterations. The adaptive step size $\alpha$ is updated as follows:

$$\alpha(t + 1) = \theta^{(t_{\text{max}})} \cdot \alpha(t)$$

(8)

where $\alpha(t)$ is the value of $\alpha$ at the current iteration, $\alpha$ ranges from 0 to 1. Dynamic step size $\alpha$ convergence curve is given in Fig. 3. From Fig. 3, it can be seen that the step is large at an early phase, and then reduces when iteration increases. It is evident that our adaptive step satisfies the convergence condition Eq. (7)

From Eq. (8), we can see that the size of $\theta$ and $\lambda$ are important factors for $\alpha$. The empirical value of $\lambda$ is 0.1. In Eq. (8), if $\lim_{t \to \infty} \alpha = 0$, the parameter $\theta$ should satisfy $0 < \theta < 1$. Different $\theta$ values will obtain different results. We adopt thirteen well-known benchmark functions from the literature (Wang et al. 2017b) to test different $\theta$ values. In this experiment, MaxIter and population size are set as 5000 and 20, respectively. The parameter $\theta$ is set as 200/1013, 500/1013, 800/1013, and 1000/1013, respectively. Table 1 lists the compared results for different $\theta$ values, where $f_{\text{m}}$ denotes the mean of best function values and $SD$ represents standard deviation. From Table 1, we can see that $\theta = 1000/1013$ obtains better solutions than those with other $\theta$ values on thirteen functions. Thus, $\theta = 1000/1013$ is considered in all the following experiments.

### 3.3 Hybridized with extremal optimization (EO) algorithm

Compared with other swarm intelligent algorithms, EO algorithm exist form is not a population but exists as a single individual. An individual can be made up of many components. For instance, if $X = (x_1, x_2, x_3)$, then $x_1, x_2$, and $x_3$ are named components of $X$. Additionally, the basic EO does not need to adjust any parameters when searching, and there is only a mutation operator. Although FA offers fast exploration and exploitation, the exploitation ability of FA can be further strengthened by EO with the merits in local search ability. However, if EO is added into FA in every generation, it will slow down the convergence speed of FA and needs more computational cost. Thus, EO is introduced into FA at INV-iteration intervals. The parameter $INV$ means the obtained global optimal solution is unchanged for $INV$-iterations. For a simple benchmark function, the $INV$ range can be larger, with a range of 50-100 being more appropriate. For complex benchmark functions, the range of $INV$ should be small, and the value range should be between 1 and 50 (Chen et al. 2010).

Due to only mutation operator in EO, it plays a vital role in the whole procedure. In this study, we use the hybrid Gaussian–Cauchy mutation operator (Chen et al. 2010). The EO introduced in FA for a minimization problem is given in Algorithm 2 (Chen et al. 2010) and Fig. 4 gives the corresponding process.
3.4 Framework of IFA-EO

The proposed IFA-EO uses MFA’s movement formula (Jr et al. 2004). In memetic FA, the movement formula is redefined. The movement formula is defined as follows (Jr et al. 2004).

\[
    x_i = x_i + \beta e^{-\gamma \epsilon_i} (x_j - x_i) + \alpha s_d \epsilon_i
\]

where \( \beta_{\text{min}} \) means the minimum value of \( \beta \), and \( s_d \) means the length scale of each designed variable. The values of \( \beta_{\text{min}} \) and \( \beta_0 \) range from 0 to 1. From Eq. (10), the range of attractiveness \( \beta \) is \( \beta_{\text{min}} \) to \( \beta_0 \).

The IFA-EO algorithm flowchart is illustrated in Fig. 5, and corresponding framework of IFA-EO is described in Algorithm 3.

**Algorithm 2: EO algorithm**

1. Obtain the i-th firefly to be mutated by EO operation;

2. For \( X_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \)

   (a) Perform the mutation operation: mutate each component in \( X_i \) one by one and keep other components unchanged.

   Then \( D \) new positions \( X_k (k = 1, 2, \ldots, D) \) can be obtained;

   (b) Evaluate the local fitness \( \delta_k = f(X_k) - f(X_i) \) of each component \( x_k, k \in \{1, \ldots, D\} \);

   (c) Rank all the fitness and find the component \( x_j \) with the worst fitness, i.e., \( \delta_j \leq \delta_k \) for all \( k \);

   (d) Select one solution \( X_j = (x_{j1}, x_{j2}, \ldots, x_{jD}, \ldots, x_{iD}) \), where \( x_j \) is the mutated component;

   (e) Accept \( X_i = X_j \) unconditionally;

3. Return \( X_i \) and \( f(X_i) \).
3.5 Computational complexity of our algorithms

According to the pseudo-code of FA (Yang 2010b), it can be found that the computational complexity of FA is $O(N^2 \cdot MaxIter)$. Compared to other intelligent optimization algorithms, including PSO, ABC, and BA, standard FA has higher complexity. In one iteration, the algorithm complexity of PSO, ABC, and BA are both $O(N)$. Since the

\begin{algorithm}
\caption{IFA-EO algorithm}
\begin{algorithmic}
\State Objective function $f(X); X = (x_1, x_2, ..., x_d)^T$
\State Define light absorption coefficient $\gamma$, initial step size $\alpha$, original light intensity $\beta_0$, minimum light intensity $\beta_{min}$
\State Define populations number of firefly $n$, the maximum number of iterations $MaxIter$
\State Define the maximum number of selective probability $P_{max}$, the minimum number of selective probability $P_{min}$
\State Generate an initial population of firefly $X_i$ ($i=1, 2, ..., n$)
\State Light intensity $I_i$ at $X_i$ is depended on $f(X_i)$
\While {($t < MaxIter$)}
\State Random generate $q \in (0,1)$
\State Calculate $P$ according to Eq. (6)
\State Calculate step size $\alpha$ according to Eq. (8)
\For {i=1 : n} //all n fireflies
\If {($q < P$)}
\For {j=1 : n} //all n fireflies (inner loop)
\If {($I_i < I_j$)}
\State Calculate $r_{ij}, \beta$ according to Eq. (3), Eq. (10)
\State Move firefly $i$ toward $j$ according to Eq. (9)
\State Assess new solution and update light intensity;
\EndIf
\EndFor
\Else
\For {j=i-1 : i}
\State $j=(j+n)\% n$
\If {($I_i < I_j$)}
\State Calculate $r_{ij}, \beta$ according to Eq. (3), Eq. (10)
\State Move firefly $i$ toward $j$ according to Eq. (9)
\State Assess new solution and update light intensity;
\Else
\State Random generate $\eta \in (0,1)$
\If {($\eta < 0.01$)}
\State Calculate $r_{ij}, \beta$ according to Eq. (3), Eq. (10)
\State Move firefly $i$ toward $j$ according to Eq. (9)
\State Assess new solution and update light intensity;
\EndIf
\EndIf
\EndFor
\EndIf
\EndFor
\EndWhile

\end{algorithmic}
\end{algorithm}
attracting model of our algorithm adopts the probability choice strategy and introduces the EO algorithm, our algorithm complexity is divided into the following situations. We only study the computational complexity at one iteration of IFA-EO.

- When using a single attraction mode and not introducing EO, the algorithm complexity is $O(N)$, where $N$ means the population size, and $D$ means the number of dimensions.
- When using a single attraction mode and introducing EO, the algorithm complexity is $O(ND)$.
- When using a full attraction mode and not introducing EO, the algorithm complexity is $O(N^2)$.
- When using a full attraction mode and introducing EO, the algorithm complexity is $O(N^2D)$.

From the above analysis, we can see the complexity of IFA-EO is $O(N)$ in the best case and $O(N^2D)$ in the worst case. From the above analysis, we can summarize as follows, compared to the FA, in the worst case, IFA-EO has a slightly higher computational complexity. But in the best case, the computational complexity of IFA-EO is very low.

Remark 1: Computational complexity is also important in the evolutionary algorithm domain. To achieve a reasonable level of optimization, the smaller computational complexity of evolutionary algorithm will save more computational resources. Also, there are many works in

| Function | $\theta = 200/1013$ | $\theta = 500/1013$ | $\theta = 800/1013$ | $\theta = 1000/1013$ |
|----------|-------------------|-------------------|-------------------|-------------------|
| $f_1$    | 5.84E-06          | 1.12E-05          | 5.37E-10          | 2.33E-09          | 2.27E-32          | 6.91E-32          | 1.10E-108         | 5.73E-108         |
| $f_2$    | 7.57E-03          | 5.81E-03          | 6.63E-04          | 6.38E-04          | 4.88E-05          | 7.95E-05          | 3.11E-42          | 9.18E-42          |
| $f_3$    | 8.43E + 02        | 3.92E + 02        | 3.86E + 02        | 1.61E + 02        | 5.32E + 01        | 4.23E + 01        | 4.93E-34          | 2.59E-33          |
| $f_4$    | 5.98E-01          | 5.01E-01          | 1.42E-02          | 1.00E-02          | 9.43E-04          | 1.14E-03          | 7.39E-35          | 3.81E-34          |
| $f_5$    | 4.45E + 01        | 2.73E + 01        | 4.48E + 01        | 2.73E + 01        | 4.16E + 01        | 2.69E + 01        | 2.80E + 01        | 1.58E + 01        |
| $f_6$    | 0.00E + 00        | 0.00E + 00        | 0.00E + 00        | 0.00E + 00        | 0.00E + 00        | 0.00E + 00        | 0.00E + 00        | 0.00E + 00        |
| $f_7$    | 5.81E-01          | 2.86E-01          | 5.56E-01          | 2.65E-01          | 5.45E-01          | 2.97E-01          | 4.45E-01          | 2.86E-01          |
| $f_8$    | 6.00E + 03        | 8.46E + 02        | 5.33E + 03        | 6.30E + 02        | 4.80E + 03        | 8.15E + 02        | 3.56E + 03        | 6.95E + 02        |
| $f_9$    | 1.08E-01          | 1.01E-01          | 1.65E-01          | 2.44E-01          | 1.42E-01          | 1.49E-01          | 1.33E + 00        | 1.01E + 00        |
| $f_{10}$ | 5.22E-03          | 1.22E-02          | 6.32E-07          | 1.05E-06          | 5.10E-14          | 1.95E-14          | 2.76E-14          | 7.13E-15          |
| $f_{11}$ | 4.87E-03          | 6.87E-03          | 2.05E-03          | 4.89E-03          | 1.31E-03          | 2.97E-03          | 2.59E-17          | 6.20E-17          |
| $f_{12}$ | 7.10E-03          | 2.59E-02          | 7.02E-08          | 1.87E-07          | 7.72E-16          | 2.87E-15          | 1.64E-32          | 2.20E-33          |
| $f_{13}$ | 9.09E-04          | 2.81E-03          | 6.09E-04          | 2.08E-03          | 2.23E-15          | 1.04E-14          | 3.13E-32          | 9.70E-33          |

**Table 1** Comparative performance for test functions $f_1$ to $f_{13}$ with different $\theta$ values

Bold face indicates the best results

Fig. 4 The process of EO
discussing in computational complexity of evolutionary algorithms, e.g., Ref. (Chen et al. 2009).

4 Experimental results and discussion

4.1 Summary of experimental conditions

In this section, we illustrate the performance of the proposed IFA-EO algorithm according to five experiments. Table 2 presents conditions of each experiment including dimension, the competitors, population size, the maximum number of fitness evaluations, and the number of runs. Five experiments are designed for different purposes. To be more specific, in the first experiment (Exp. 1), the aim is to verify that the proposed IFA-EO as a kind of FA variant can realize better performance than other FA variants. Thus, seven FA variants, i.e., ApFA (Wang et al. 2017a), NaFA (Wang et al. 2017b), VSSFA (Yu et al. 2015), WSSFA (Yu et al. 2014), CFA (Gandomi et al. 2013), MFA (Jr et al. 1204) and FA (Yang 2010b) are viewed as the competitors. Considering the IFA-EO is efficiently using EO mechanism, EO-based algorithms are needed to compare. Here, we choose three recently related methods, i.e., PEO (Chen et al. 2006), PSO-EO (Chen et al. 2010), and RPEO-PLM (Li et al. 2016) as comparisons in the second experiment (Exp. 2). To further prove the performance of IFA-EO, the third experiment is devoted to comparing IFA-EO algorithm with other two well-recognized algorithms. These two competitors are AD-IFA (Wu et al. 2020) and L-HSADE (Biswas et al. 2019). In the fourth experiment (Exp. 4), FA with different modified strategies is shown to demonstrate the performance of different strategies. In addition, to investigate the impact of population size on the performance of IFA-EO, we design the fifth experiment (Exp. 5).

It is worth mentioning that different adjustable parameters may influence the performance of IFA-EO. Here, the related parameters are determined by the trial-and-error method. In all the following experiments, the initial parameters $\lambda$, $\alpha$, $\gamma$, $\beta_0$, and $\beta_{\min}$ are set as 0.1, 0.2, 1.0, 1.0 and 0.2, respectively. The minimum and maximum attraction probability $P_{\min}$, $P_{\max}$ are set to 0.05 and 0.8, respectively. The considered functions are all minimized problems. Experiments 1–5 of IFA-EO are performed in JAVA software on a 3.20 GHz computer with processor i5-6500U and 8 GB RAM.

4.2 Exp. 1 Comparison with FA variants

This group of well-known benchmark functions shown in Table 3 is selected from the literature (Wang et al. 2017b). For these test functions in Table 3, since the $f_{1}, f_{2}, f_{5}, f_{6}, f_{10}, f_{12}, f_{13}$ are simple test functions, the firefly procedures can find a better solution, so when the optimal solution does not change for 500 generations, we introduce EO. For $f_{3}$, the value of $INV$ is 100. For complex test functions, the $f_{11}$, we set $INV$ as 10 and the $f_{7}$, $INV = 50$. Since the $f_{3}, f_{5}, f_{9}$ function is easy to trap into the local optimums, we introduce EO when the optimal solution does not change in five generations. Recently, some FA variants, e.g., ApFA (Wang et al. 2017a), NaFA (Wang et al. 2017b), VSSFA (Yu et al. 2015), CFA (Gandomi et al. 2013), MFA (Jr et al. 1204) and FA (Yang 2010b) have been proposed. Due to the improvement on standard FA, these FA variants have been proved their performance through various optimization problems. The main framework of IFA-EO algorithm is FA, which can be viewed as another FA variant. Thus, it is necessary to compare with the above seven FA variants to verify that IFA-EO is another efficient FA variant. Table 4 presents average results $f_{\text{avg}}$ of these eight FA variants for thirteen benchmark test functions on 30 run times. The experimental results of
algorithms FA, VSSFA, WSSFA, MFA, CFA, and NaFA are excerpted from the literature (Wang et al. 2017b). The results of ApFA are taken from reference (Wang et al. 2017a). Note that bold values in the table of this paper mean the best unless otherwise stated. For Table 5, it can be seen that IFA-EO algorithm performs the best in 8 out of 13 test functions \( f_1 \)-\( f_{13} \), the same as \( f_6 \) for MFA, CFA, NaFA, and ApFA, and the same as NaFA for function \( f_{11} \). For some low-dimensional problems, VSSFA and WSSFA have good performance (Yu et al. 2015, 2014), but they fail to obtain well solutions compared to other FA variants on 30-dimensional problems. The comparison results of IFA-EO and the other seven FA variants are denoted by \( w/t/l \), which indicates that IFA-EO has better performance in \( w \) functions, similar performance in \( t \) functions, and worse performance in \( l \) functions. Thus, compared to the other seven competitors, IFA-EO can achieve almost the same or better performance on the considered functions \( f_1 \)-\( f_{13} \).

In addition, we use two nonparametric statistic tests (Derrac et al. 2011), i.e., Friedman test and Quade test to further compare all eight FA variants and the Bonferroni–Dunn method is selected as the post-hoc test performed by the KEEL (Alcalá-Fdez et al. 2009). Table 5 presents ranks, statistics, and \( p \)-values achieved by two nonparametric statistic tests for IFA-EO and other seven FAs. It is obvious that IFA-EO ranks the first in Quade tests with a level of significance \( \alpha = 0.05 \). NaFA and IFA-EO obtains similar performance in terms of Friedman test. Note that IFA-EO achieves better performance than NaFA in terms of \( f_m \).

Remark 2: Although the value 1.10E-108 and 1.64E-32 in Table 4 is very close, any improved performance may mean the improvements of exploration ability and exploitation ability. For example, Fig. 6 gives an example. The term “A” means the global optimum and the term “B” means the local optimum. In some real applications, it is difficult to know how many local optimums in the optimization function. If the algorithm can find “A” rather than “B”, the algorithm may have good exploration ability. Similarly, the term “C” is close to global optimum, but is fails to find the global optimum “A”. If the algorithm can find “A” rather than “C”, the algorithm may have good exploitation ability. Due to lack of knowledge of real-world optimization function, we cannot ensure every “B” or “C”
Table 3 Benchmark functions $f_1$-$f_{13}$ with $D = 30$

| Problem | Function expression | Search space | Global minimum |
|---------|----------------------|--------------|----------------|
| $f_1$   | $f_1(x) = \frac{D}{\prod_{i=1}^{D} x_i}$ | $[-100,100]$ | 0              |
| $f_2$   | $f_2(x) = \frac{D}{\prod_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|}$ | $[-10,10]$ | 0              |
| $f_3$   | $f_3(x) = \frac{D}{\prod_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2}$ | $[-100,100]$ | 0              |
| $f_4$   | $f_4(x) = \max\{|x_i|, 1 \leq i \leq D\}$ | $[-100,100]$ | 0              |
| $f_5$   | $f_5(x) = \frac{D}{\prod_{i=1}^{D} (100 (x_i^2 - x_{i+1}) + (x_i - 1)^2)}$ | $[-30,30]$ | 0              |
| $f_6$   | $f_6(x) = \sum_{i=1}^{D} (|x_i + 0.5|)^2$ | $[-100,100]$ | 0              |
| $f_7$   | $f_7(x) = \sum_{i=1}^{D} |x_i|^4 + \text{random}[0,1]$ | $[-1.28,1.28]$ | 0              |
| $f_8$   | $f_8(x) = 418.9829 \cdot D - \prod_{i=1}^{D} x_i \sin(\sqrt{|x_i|})$ | $[-500,500]$ | 0              |
| $f_9$   | $f_9(x) = \frac{D}{\prod_{i=1}^{D} (x_i^2 - 10 \cos 2\pi x_i + 10)}$ | $[-5.12,5.12]$ | 0              |
| $f_{10}$| $f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^{D} \cos (2\pi x_i)\right) + 20 + \epsilon$ | $[-32.32]$ | 0              |
| $f_{11}$| $f_{11}(x) = \frac{D}{\prod_{i=1}^{D} \sqrt{\sin^2(3\pi x_i)}} - \frac{1}{D} \prod_{i=1}^{D} \cos \left(\frac{x_i}{\sqrt{D}}\right) + 1$ | $[-600,600]$ | 0              |
| $f_{12}$| $f_{12} = \tfrac{D}{\prod_{i=1}^{D}} (y_0 - 1)^2 + 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + \sin^2(\pi y_{i+1})]$ | $[-50,50]$ | 0              |
|         | $+ \sum_{i=1}^{D} u(x_i, 10, 100, 4), y_i = 1 + \frac{x_i + 1}{4}$ | $[-50,50]$ | 0              |
|         | $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$ | $[-50,50]$ | 0              |
| $f_{13}$| $f_{13} = 0.1 \{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})]$ | $[-50,50]$ | 0              |
|         | $+ (x_0 - 1)^2 [1 + \sin^2(2\pi x_0)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$ | $[-50,50]$ | 0              |
is very close to "A". Thus, we report these errors. For some real-world cases, below a threshold (e.g., 1.0E-8) the error can be considered as zero. In this paper, we test the exploration ability and exploitation ability of IFA-EO, thus the errors below 1.0E-8 are not considered as zero.

4.3 Exp. 2: Comparison with EO-based algorithms

This group of well-known benchmark functions shown in Table 6 is selected from the literature (Wang et al. 2017b). In this experiment, we choose three recently published related EO variants, i.e., PEO (Chen et al. 2006), PSO-EO (Chen et al. 2010), and RPEO-PLM (Li et al. 2016) as comparisons to demonstrate the effectiveness of combination of EO mechanism in FA. The experimental results of four algorithms, i.e., IFA-EO, PEO, PSO-EO, and RPEO-PLM for eight test functions are presented in Table 7. $f_m$ represents the average results of run 30 times. The results of RPEO-PLM are taken from the reference (Li et al. 2016). Table 7 implies that the proposed IFA-EO algorithm performs the best in 4 out of 8 test functions $h_1$-$h_8$, the

![Fig. 6 Two cases of small error](image-url)
same as PSO-EO for functions \( h_2 \) and \( h_7 \), and the same as RPEO-PLM for function \( h_7 \). Therefore, IFA-EO algorithm has a more powerful ability to solve such high-dimension test functions \( h_1-\ldots-h_8 \) than these three compared algorithms.

To further compare four EO-based swarm intelligence algorithms, we use KEEL (Alcalá-Fdez et al. 2009) software to conduct the Friedman test and Quade test and the Bonferroni–Dunn method is selected as the post-hoc test. Table 8 gives the rankings obtained by IFA-EO and other EO-based algorithms. Based on the Friedman rankings, IFA-EO achieves the best performance. We can see that IFA-EO is obviously better than PEO and is significantly better than RPEO-PLM. Compare IFA-EO with PSO-EO, they achieve similar results.

### Table 6 Benchmark functions \( h_1-\ldots-h_8 \) with \( D = 100 \)

| Problem | Function expression | Search space | Global minimum |
|---------|---------------------|--------------|----------------|
| \( h_1 \) | \( h_1 = \sum_{i=1}^{D} \frac{\sin(10x_i)}{10x_i} \) | \([-0.5,0.5]\) | 0 |
| \( h_2 \) | \( h_2 = \sum_{i=1}^{D} \left[x_i + 0.5\right]^2 \) | \([-100,100]\) | 0 |
| \( h_3 \) | \( h_3 = \sum_{i=1}^{D} \left[x_i^2 - 10 \cos(2\pi x_i) + 10\right] \) | \([-5.12,5.12]\) | 0 |
| \( h_4 \) | \( h_4 = \sum_{i=1}^{D} x_i^2 \) | \([-5.12,5.12]\) | 0 |
| \( h_5 \) | \( h_5 = -20 \exp \left(-0.02 \sqrt{\sum_{i=1}^{D} x_i^2}\right) - \exp \left(\frac{\sum_{i=1}^{D} \cos(2\pi x_i)}{D}\right) + 20 + e \) | \([-30,30]\) | 0 |
| \( h_6 \) | \( h_6 = 418.9828D - \sum_{i=1}^{D} x_i \sin(\sqrt{|x_i|}) \) | \([-500,500]\) | 0 |
| \( h_7 \) | \( h_7 = 6D + \sum_{i=1}^{D} \left|x_i\right| \) | \([-5.12,5.12]\) | 0 |
| \( h_8 \) | \( h_8 = 1 + \frac{1}{\prod_{i=1}^{D} \cos \left(\frac{x_i}{\sqrt{D}}\right)} \) | \([-600,600]\) | 0 |

### Table 7 Comparative results on test functions \( h_1-\ldots-h_8 \) with \( D = 100 \)

| Function | IFA-EO | PEO (Chen et al. 2006) | RPEO-PLM (Li et al. 2016) | PSO-EO (Chen et al. 2010) |
|----------|--------|------------------------|---------------------------|--------------------------|
| Mean     | Rank   | Mean                   | Rank                      | Mean                     | Rank                     |
| \( h_1 \) | 3.53E-07 | 2 | 2.62E-03 | 4 | 1.89E-06 | 3 | 0.00E + 00 | 1 |
| \( h_2 \) | **0.00E + 00** | 1 | 1.62E-06 | 3 | 1.66E-05 | 4 | 0.00E + 00 | 1 |
| \( h_3 \) | 6.85E-05 | 3 | 1.26E + 01 | 4 | 2.76E-06 | 2 | **0.00E + 00** | 1 |
| \( h_4 \) | **1.62E-111** | 1 | 3.60E-10 | 3 | 1.39E-08 | 4 | 2.15E-11 | 2 |
| \( h_5 \) | **1.07E-13** | 1 | 2.04E + 01 | 4 | 6.98E-08 | 2 | 5.02E-05 | 3 |
| \( h_6 \) | 8.01E + 03 | 3 | 7.52E + 04 | 4 | **1.38E-03** | 1 | 8.62E + 00 | 2 |
| \( h_7 \) | **0.00E + 00** | 1 | 3.02E + 00 | 4 | **0.00E + 00** | 1 | **0.00E + 00** | 1 |
| \( h_8 \) | 6.62E-16 | 2 | **0.00E + 00** | 1 | 5.01E-06 | 3 | 9.27E-04 | 4 |
| Average rank | **1.75** | 3.375 | 2.5 | **1.875** | |
| Final rank | **1** | 4 | 3 | 2 | |

Bold face indicates the best results
4.4 Exp. 3: Comparison with L-SHADE and AD-IFA

In Exp. 3, the performance of the IFA-EO is compared with recently improved FA, called an adaptive logarithmic spiral-levy FA (AD-IFA) and advanced DE variant, called linear population size reduction technique of success history based adaptive differential evolution (L-SHADE).

| Test function | Optimum | Algorithm   | \(f_m\) (rank) | SD (rank) | Final rank |
|---------------|---------|-------------|----------------|-----------|------------|
| \(f_1\) 0 (min) | IFA-EO | 5.68E-09(2) | 1.02E-08(3) | 2         |
|              | AD-IFA | 3.91E-08(3) | 9.42E-09(2) | 3         |
|              | L-SHADE| \textbf{4.87E-23(1)} | \textbf{2.46E-22(1)} | 1         |
| \(f_2\) 0 (min) | IFA-EO | 4.58E-04(2) | 1.26E-03(3) | 2         |
|              | AD-IFA | 2.00E-03(3) | 5.78E-04(2) | 3         |
|              | L-SHADE| \textbf{2.74E-21(1)} | \textbf{9.60E-21(1)} | 1         |
| \(f_3\) 0 (min) | IFA-EO | \textbf{7.77E-04(1)} | \textbf{3.21E-03(1)} | 1         |
|              | AD-IFA | 6.22E + 00(3) | 8.88E + 00(3) | 3         |
|              | L-SHADE| 2.29E-02(2) | 5.56E-02(2) | 2         |
| \(f_4\) 0 (min) | IFA-EO | \textbf{7.12E-02(1)} | \textbf{5.18E-02(1)} | 1         |
|              | AD-IFA | 1.43E + 01(2) | 5.97E + 00(3) | 2         |
|              | L-SHADE| 3.18E + 01(3) | 4.12E + 00(2) | 3         |
| \(f_5\) 0 (min) | IFA-EO | 4.45E-01(2) | 2.86E-01(3) | 2         |
|              | AD-IFA | 5.45E-01(3) | 1.60E-01(2) | 3         |
|              | L-SHADE| \textbf{8.77E + 01(1)} | \textbf{4.21E + 01(1)} | 1         |
| \(f_6\) 0 (min) | IFA-EO | 2.27E-08(2) | 5.39E-08(3) | 2         |
|              | AD-IFA | 4.15E-08(3) | 1.26E-08(2) | 3         |
|              | L-SHADE| \textbf{3.53E-22(1)} | \textbf{1.30E-21(1)} | 1         |
| \(f_7\) 0 (min) | IFA-EO | \textbf{4.30E-08(1)} | \textbf{1.57E-07(1)} | 1         |
|              | AD-IFA | 9.02E + 01(3) | 2.36E + 01(3) | 3         |
|              | L-SHADE| 9.62E-02(1) | 1.47E + 00(2) | 2         |
| \(f_8\) −20,949 (min) | IFA-EO | \textbf{-1.64E + 04(2)} | \textbf{8.01E + 02(2)} | 2         |
|              | AD-IFA | \textbf{-1.33E + 04(3)} | \textbf{1.35E + 03(3)} | 3         |
|              | L-SHADE| \textbf{-2.03E + 04(1)} | \textbf{2.86E + 02(1)} | 1         |
| \(f_9\) 0 (min) | IFA-EO | \textbf{4.30E-08(1)} | \textbf{1.57E-07(1)} | 1         |
|              | AD-IFA | 9.02E + 01(3) | 2.36E + 01(3) | 3         |
|              | L-SHADE| 9.62E-02(1) | 1.47E + 00(2) | 2         |
| \(f_{10}\) 0 (min) | IFA-EO | \textbf{1.66E-05(1)} | \textbf{1.63E-05(1)} | 1         |
|              | AD-IFA | 9.73E-01(2) | 1.02E + 00(2) | 2         |
|              | L-SHADE| 7.88E + 00(3) | 1.16E + 00(3) | 3         |
| \(f_{11}\) 0 (min) | IFA-EO | \textbf{5.09E-03(1)} | \textbf{6.75E-03(1)} | 1         |
|              | AD-IFA | 6.89E-03(2) | 1.02E-02(2) | 2         |
|              | L-SHADE| 1.04E-01(3) | 1.58E-01(3) | 3         |
| \(f_{12}\) 0 (min) | IFA-EO | \textbf{1.64E-32(1)} | \textbf{2.20E-33(1)} | 1         |
|              | AD-IFA | 2.77E + 00(3) | 1.59E + 00(2) | 3         |
|              | L-SHADE| 2.20E + 00(2) | 2.21E + 00(3) | 2         |
| \(f_{13}\) 0 (min) | IFA-EO | \textbf{9.70E-33(1)} | \textbf{3.13E-32(1)} | 1         |
|              | AD-IFA | 2.50E + 00(3) | 9.63 + E00(3) | 3         |
|              | L-SHADE| 1.39E + 00(2) | 2.44E + 00(2) | 2         |

Bold face indicates the best results

AD-IFA uses the logarithmic-spiral guidance the paths of fireflies and considers an adaptive switching to keep the suitable balance between exploration ability and exploitation ability. The effectiveness of AD-IFA has been illustrated by solving various test functions and real-life engineering problems. Thus, as one of the improved version of FA, the performance of IFA-EO can be further testified by comparing with this state-of-the-art FA.
Algorithm Friedman ranking Quade ranking

IFA-EO 1.4615 1.3736
AD-IFA (Wu et al. 2020) 2.7692 2.7363
L-SHADE (Biswas et al. 2019) 1.7692 1.8901

Statistic 12.15 6.99

p-value 0.002295 0.00405

Bold face indicates the best results

Table 11 Comparative results for test functions \( f_1 \)–\( f_{13} \) with \( D = 30 \)

| Function | IFA-EO | FA-PA | FA-A | FA-EO | FA-B |
|----------|--------|-------|------|-------|------|
| \( f_1 \) | 3.81E-14 | 1.95E-14 | 2.97E-04 | 2.76E-02 | 4.08E-02 |
| \( f_2 \) | 4.16E-10 | 1.67E-05 | 8.12E-04 | 5.30E-02 | 6.66E-02 |
| \( f_3 \) | 5.66E-11 | 7.46E-10 | 7.49E-04 | 5.10E-02 | 5.49E-02 |
| \( f_4 \) | 8.81E-06 | 1.24E-06 | 7.66E-03 | 4.66E-01 | 9.27E-01 |
| \( f_5 \) | 1.07E-03 | 3.48E-03 | 1.46E-02 | 8.47E-01 | 1.08E+00 |
| \( f_6 \) | 2.40E-04 | 3.90E-04 | 1.15E-02 | 6.57E-01 | 1.01E+00 |
| \( f_7 \) | 1.24E + 01 | 1.86E + 01 | 2.68E + 00 | 1.83E-01 | 1.54E-01 |
| \( f_8 \) | 2.10E + 02 | 3.35E + 02 | 9.62E + 01 | 1.49E + 00 | 2.05E+00 |
| \( f_9 \) | 8.02E + 01 | 1.14E + 02 | 2.51E + 01 | 6.36E-01 | 5.34E+00 |
| \( f_{10} \) | 2.27E-04 | 4.66E-04 | 1.89E-02 | 1.30E+00 | 6.66E-01 |
| \( f_{11} \) | 3.69E-03 | 2.58E-03 | 3.70E-02 | 1.01E+01 | 1.03E+01 |
| \( f_{12} \) | 1.18E-03 | 1.23E-03 | 2.90E-02 | 5.00E-00 | 5.34E+00 |
| \( f_{13} \) | 5.86E-03 | 2.58E-01 | 2.52E-01 | 2.81E+01 | 2.95E+01 |

Algorithm Friedman ranking Quade ranking

Furthermore, as an advanced DE variant, L-SHADE is considered as the competitor to show the performance of IFA-EO. The adjustable parameters of AD-IFA and L-SHADE are suggested by the corresponding references. The thirteen test functions are given in Table 3 with \( D = 50 \). The \( NP \) is set as 20 and the \( FES_{\text{max}} \) is set as 300,000 as the stop criterion. Other parameters are the same as suggested (Wu et al. 2020; Biswas et al. 2019). Table 9 presents the comparative performances of IFA-EO algorithm.
with above two algorithms on test functions $f_1$-$f_{13}$ in terms of $f_m$ and SD. Symbol “(1)” $\sim$ “(5)” implies the rank of algorithms in terms of $f_m$ and SD achieved from 30 independent runs. In addition, Table 10 gives the Friedman and Quade rankings. From Tables 9 and 10, we can see the following observation:

(1) Through comparing IFA-EO with AD-IFA, it can be clearly observed that IFA-EO achieves all better performance than AD-IFA in solving $f_1$-$f_{13}$ in terms of $f_m$ and SD except the SD of $f_1$ and $f_2$.

(2) By comparing IFA-EO and L-SHADE, it is clear that IFA-EO wins in seven test functions. Also, L-SHADE achieves better performance than IFA-EO in other six functions. They achieve similar performance for $f_1$-$f_{13}$.

(3) IFA-EO obtains better ranks than AD-IFA and has similar performance with L-SHADE in the view of nonparametric statistics test.

L-SHADE is an improved version of DE and achieves better performance than various DE algorithms. Although the proposed IFA-EO achieves similar performance with L-SHADE, it significantly outperforms the recently improved version of FA, i.e., AD-IFA (Wu et al. 2020). Thus, as one kind of FA variant, IFA-EO may be considered as a potential algorithm in evolutionary algorithms. From Table 9, it can be found the L-SHADE performs worse performance in $f_4$, $f_9$, $f_{10}$, $f_{12}$, and $f_{13}$ while IFA-EO performs well in solving these functions. Thus, the improvements in IFA-EO may further enhance the performance of L-SHADE.

### 4.5 Exp. 4: Effects of different strategies

From the above first three experiments, we have shown the superiority of the proposed IFA-EO algorithm for two well-known benchmark functions. As described in Sect. 3, IFA-EO employs three strategies: a probability choice for combination of full attraction model and the single attraction model, and acceptance of some worse solutions through small probability in the single attraction model; an

| Algorithm | Friedman ranking | Quade ranking |
|-----------|------------------|---------------|
| IFA-EO    | 1.6154           | 1.7692        |
| FA-PA     | 2.7308           | 2.9505        |
| FA-A      | 3.0385           | 2.9945        |
| FA-EO     | 3.2308           | 3.2088        |
| FA-B      | 4.3846           | 4.0769        |
| Statistic | 20.6             | 3.2207        |
| p-value   | 0.00038          | 0.02021       |

Bold face indicates the best results

Fig. 7 Convergence curves of different FAs for $f_1$ test function
Fig. 8 Convergence curves of different FAs for $f_2$ test function

Fig. 9 Convergence curves of different FAs for $f_9$ test function
Fig. 10 Convergence curves of different FAs for $f_{10}$ test function

Fig. 11 Convergence curves of different FAs for $f_{12}$ test function
adaptive step with iterations; and the combination of EO procedure. To assess the performance of these three strategies, we compare five FAs equipped with different strategies in the fourth experiment to verify the influence on the performance of the IFA-EO algorithm. The considered algorithms are given as follows.

- FA + Beta (termed as FA-B)
- FA + Beta + adaptive step (termed as FA-A)
- FA + Beta + EO procedure (termed as FA-EO)
- FA + Beta + probability choice + adaptive step (termed as FA-PA)
- FA + Beta + probability choice + adaptive step + EO procedure (i.e., IFA-EO)

In this experiment, all algorithms use the same parameter settings as described in Sect. 4.1. And INV set as 5 in the FA-EO for all test functions.

Table 11 gives the competitive results of compared five algorithms. Table 12 lists the mean ranks and the results of Friedman and Quade tests, respectively. In addition, Figs. 7, 8, 9, 10, 11 and 12 show the convergence curves of FA-B, FA-A, FA-EO, FA-PA, and IFA-EO on the selected functions. From Tables 11 and 12 and Figs. 7, 8, 9, 10, 11 and 12, we can see the following observations:

(a) Through comparing IFA-EO with FA-PA, it can be clearly observed that combination of EO procedure can improve the accuracy of FA-B.

(b) By comparing FA-B with FA-A, FA-PA and IFA-EO, it can be found that the three considered

Table 13 Result achieved by IFA-EO under different population sizes

| Function | IFA-EO |
|----------|--------|
|          | 10     | 20     | 30     | 40     |
| $f_1$    | 1.38E-96 | 1.10E-108 | 2.77E-15 | 6.42E-02 |
| $f_2$    | 6.67E-06 | 3.11E-42 | 1.97E-08 | 1.31E-01 |
| $f_3$    | 3.45E-01 | 4.93E-34 | 6.13E-05 | 3.88E+01 |
| $f_4$    | 1.49E-05 | 7.39E-35 | 2.52E-07 | 1.32E-01 |
| $f_5$    | 3.75E+01 | 2.80E+01 | 3.69E+01 | 3.39E+01 |
| $f_6$    | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| $f_7$    | 5.57E-01 | 4.45E-01 | 5.69E-01 | 5.67E-01 |
| $f_8$    | 2.17E+03 | 3.56E+03 | 3.47E+03 | 3.53E+03 |
| $f_9$    | 4.74E-02 | 1.33E+00 | 2.12E+01 | 1.61E+01 |
| $f_{10}$ | 4.80E-14 | 2.76E-14 | 5.30E-09 | 8.84E-02 |
| $f_{11}$ | 1.40E-03 | 2.59E-17 | 1.10E-13 | 1.75E-01 |
| $f_{12}$ | 3.63E-06 | 1.64E-32 | 1.35E-15 | 3.22E-04 |
| $f_{13}$ | 2.42E-06 | 3.13E-32 | 1.48E-14 | 4.35E-03 |

Bold face indicates the best results.
strategies play a vital role in improving the performance of FA-B. The adaptive step strategy can significantly improve FA’s performance, combination with probability choice and adaptive step strategy or hybrid three strategies can better enhance the ability of FA-B.

(c) From Tables 11 and 12, IFA-EO achieves the best rank in terms of Friedman test and Quade test and is significantly better than other four compared algorithms.

(d) From Figs. 7, 8, 9, 10, 11 and 12, we can see that the accuracy of IFA-EO and FA-PA methods is higher than other different strategies. And IFA-EO is better than FA-PA method on the whole. In addition, the convergences of IFA-EO and FA-PA are faster than other compared methods.

Overall, each strategy has different impacts and plays a vital role in the algorithm, and IFA-EO can be considered the best method among the all different strategies in terms of Friedman and Quade tests, convergence speed, and global minimum value.

4.6 Exp. 5: Effects of population size

This experiment is devoted to investigating the impacts of population size on the performance of our algorithm IFA-EO. The population size is set to 10, 20, 30, and 40, respectively, while other adjustable parameters are the same as those in previous experiments.

Table 13 gives the mean best fitness values obtained by IFA-EO under different population sizes. It can see that the population size has a large impact on the precision of the final solutions on test functions. When the population size is set 20, IFA-EO can achieve better comprehensive solutions than other population size settings. Thus, in FA, we usually set the population size as 20.

5 Conclusions and future work

In this paper, we present an improved firefly algorithm, namely IFA-EO, which uses three improved strategies. First of all, we use the probability choice strategy to select the attractive model of firefly, which can be divided into two models: full attraction and single attraction. Full attraction needs to be compared with all fireflies, which indicates the convergence speed is slow and it takes more time. The single attraction mode is fast, but easy to fall into local optimum. Therefore, these two models are combined by the probability formula to achieve complementarity, and in the single attraction mode, the small probability accepts the worse solution. This method can dynamically balance exploration and exploitation, which speeds up convergence and saves time. Second, instead of using a fixed step size, this paper uses an adaptive step size related to the number of iterations. The value of the previous step is relatively large and is almost zero near the end. Third, EO algorithm is considered in this paper to enhance the local search ability, and solution accuracy. When the optimal solution is unchanged continuously for INV-iterations remain unchanged, EO algorithm is introduced. In the experimental part, we use two group well-known test functions to compare IFA-EO with the other thirteen algorithms. It can be seen that the IFA-EO algorithm has better performance in terms of convergence speed, solution accuracy and statistical tests. In future research work, we will try to apply the modified algorithm to some constrained continuous optimization problems and multi-objective combination optimization problems, binary optimization problems, and fuzzy clustering problems Ozsoydan 2019b; Baykasoğlu et al. 2019.

Acknowledgements This work was supported by National Natural Science Foundation of China (Nos. 61872153 and 61972288), National Science Foundation of Guangdong Province (No.2018A030313318), and Key-Area Research and Development Program of Guangdong Province (No. 2020B0101090004).

Funding Key-Area Research and Development Program of Guangdong Province (CN),2020B0101090004, Guo-Qiang Zeng, National Natural Science Foundation of China (CN),61872153, Min-Rong Chen, 61972288,Guo-Qiang Zeng, National Science Foundation of Guangdong Province (CN), 2018A030313318, Min-Rong Chen.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest Min-Rong Chen declares that she has no conflict of interest. Liu-Qing Yang declares that she has no conflict of interest. Guo-Qiang Zeng declares that he has no conflict of interest. Kang-Di Lu declares that he has no conflict of interest. Yi-Yuan Huang declares that she has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Alcalá-Fdez J, Sanchez L, Garcia S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC (2009) KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318
Alweshah M, Abdullah S (2015) Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems. Appl Soft Comput 35:513–524
Baykasoğlu A, Gölcük İ, Özsoydan FB (2019) Improving fuzzy c-means clustering via quantum-enhanced weighted superposition attraction algorithm. Hacet J Math Stat 48(3):859–882
Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems, Robots and biological systems: towards a new bionics? Springer, Berlin, Heidelberg, pp 703–712
Bhusan P, Pillai SS (2013) “Particle swarm optimization and firefly algorithm: performance analysis,” In: IEEE International Advanced Computing Conference (IACC). IEEE, pp. 746–751
Biswas PP, Suganthan PN, Wu G, Amaratunga GAJ (2019) Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm. Renew Energy 132:425–438
Boettcher S, Percus AG (1999) “Extremal optimization: methods derived from co-evolution,” In: Proceedings of the Genetic and Evolutionary Computation Conference, pp.825–832
Boettcher S, Percus AG (2000) Nature’s way of optimizing. Artif Intell 119:275–286
Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, NewYork
Chen T, He J, Sun G, Chen G, Yao X (2009) “A new approach for analyzing average time complexity of population-based evolutionary algorithms on unimodal problems”, IEEE Trans Syst, Man, and Cybern B (cybernetics) 39(5):1092–1106
Chen MR, Li X, Zhang X, Lu YZ (2010) A novel particle swarm optimizer hybridized with extremal optimization. Appl Soft Comput 10(2):367–373
Chen MR, Chen JH, Zeng GQ, Lu KD, Jiang XF (2019) An improved extremal optimization and Boltzmann Selection probability. Swarm Evol Comput 49:158–177
Chen MR, Lu YZ, Yang GK (2006) Population-based extremal optimization with adaptive Lévy mutation for constrained optimization. In: Proceedings of 2006 International Conference on Computational Intelligence and Security (CIS’06), pp. 258–261
Deb K, Pratap A, Agarwal S (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18
Fister I, Yang XS, Brest J, Fister I (2013) Jr, “Modified firefly algorithm using quaternion representation.” Expert Syst Appl 40(18):7220–7230
Gandomi AH, Yang XS, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98
Gölcük İ, Özsoydan FB (2020) Evolutionary and adaptive inheritance enhanced Grey Wolf Optimization algorithm for binary domains. Knowl-Based Syst 194:105586
Ho SY, Shu LS, Chen JH (2004) Intelligent evolutionary algorithms for large parameter optimization problems. IEEE Trans Evol Comput 8(6):522–541
Horng MH (2012) Vector quantization using the firefly algorithm for image compression. Expert Syst Appl 39(1):1078–1091
Hian H, Chen X, Wu D (2018) A switch-mode firefly algorithm for global optimization. IEEE Access 6:54177–54184
Fister Jr I, Yang XS, Fister I, Brest J (2012) “Memetic firefly algorithm for combinatorial optimization,” arXiv preprint arXiv: 1204.5165
Kazem A, Sharifi E, Hussain FK, Saberi M, Hussain OK (2013) Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Appl Soft Comput 13(2):947–958
Kennedy J (2010) Particle swarm optimization. Encycl Mach Learn. 760:766
Li LM, Lu KD, Zeng GQ, Wu L, Chen MR (2016) A novel real-coded population-based extremal optimization algorithm with polynomial mutation: a non-parametric statistical study on continuous optimization problems. Neurocomputing 174:577–587
Lu KD, Zhou WN, Zeng GQ, Zheng YY (2019) Constrained population extremal optimization-based robust load frequency control of multi-area interconnected power system. Int J Electr Power Energ Syst 105:249–271
Lv SX, Zeng YR, Wang L (2018) An effective fruit fly optimization algorithm with hybrid information exchange and its applications. Int J Mach Learn Cybern 9(10):1623–1648
Olamaei J, Moradi M, Kaboodi T (2013) “A new adaptive modified firefly algorithm to solve optimal capacitor placement problem,” In: Proceedings of the 2013 18th Conference on Electrical Power Distribution Networks (EPDC), pp. 1–6
Özsoydan FB (2019a) Effects of dominant wolves in grey wolf optimization algorithm. Appl Soft Comput 83:105658
Özsoydan FB (2019b) Artificial search agents with cognitive intelligence for binary optimization problems. Comput Ind Eng 136:18–30
Özsoydan FB, Baykasoglu A (2019) Analysing the effects of various switching probability characteristics in flower pollination algorithm for solving unconstrained function minimization problems. Neural Comput Appl 31(11):7805–7819
Özsoydan FB, Baykasoglu A (2019) Quantum firefly swarms for multimodal dynamic optimization problems. Expert Syst Appl 115:189–199
Özsoydan FB, Gölcük İ (2021) Cuckoo search algorithm with various walks. Applications of Cuckoo Search Algorithm and its Variants. Springer, Singapore, pp 47–77
Özsoydan FB, Baykasoglu A (2015) “A multi-population firefly algorithm for dynamic optimization problems.” In: IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS). IEEE, 1–7
Pan X, Xue L, Li R (2018) A new and efficient firefly algorithm for numerical optimization problems. Neural Comput Appl 31:1–9
Patle BK, Pandey A, Jagadesh A, Parhi DR (2018) Path planning in uncertain environment by using firefly algorithm. Def Technol 14(6):691–701
Storm R, Price K (1997) Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
Tang PH, Tseng MH (2013) Adaptive directed mutation for real-coded genetic algorithms. Appl Soft Comput 13:600–614
Trunfio GA (2014) Enhancing the firefly algorithm through a cooperative coevolutionary approach: an empirical study on benchmark optimisation problems. Int J Bio-Inspir Comput 6(2):108–125
Wang H, Wang W, Sun H, Rahaman May S (2016) Firefly algorithm with random attraction. Int J Bio-Inspir Comput 8(1):33–41
Wang H, Zhou X, Sun H, Yu X, Zhao J, Cui L (2017a) Firefly algorithm with adaptive control parameters. Soft Comput 21(17):5091–5102
Wang H, Wang W, Zhou X, Sun H, Zhao J, Yu X, Cui Z. (2017b) Firefly algorithm with neighborhood attraction. Inf Sci 438:95–106
Wang H, Wang W, Cui Z, Zhou X, Zhao J, Li Y (2018) A new dynamic firefly algorithm for demand estimation of water resources. Inf Sci 438:95–106
Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evolut Comput 1:67–82
Wu J, Wang YG, Burrage K, Tian YC, Lawson B, Ding Z (2020) An improved firefly algorithm for global continuous optimization problems. Exp Syst Appl 149:113340

Springer
Yang XS (2010a) A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization, Berlin, Heidelberg, pp 65–74
Yang XS, “Nature-inspired metaheuristic algorithms,” Luniver press, 2010b.
Yu S, Su S, Lu Q, Huang L (2014) A novel wise step strategy for firefly algorithm. Int J Comput Math 91(12):2507–2513
Yu S, Zhu S, Ma Y, Mao D (2015) A variable step size firefly algorithm for numerical optimization. Appl Math Comput 263:214–220
Zeng GQ, Chen J, Dai YX, Li LM, Zheng CW, Chen MR (2015) Design of fractional order PID controller for automatic regulator voltage system based on multi-objective extremal optimization. Neurocomputing 160:173–184
Zeng GQ, Xie XQ, Chen MR, Weng J (2019) Adaptive population extremal optimization-based PID neural network for multi-variable nonlinear control systems. Swarm Evol Comput 44:320–334
Zhao C, Wu C, Chai J, Wang X, Yang X, Lee J, Kim MJ (2017) Decomposition-based multi-objective firefly algorithm for RFID network planning with uncertainty. Appl Soft Comput 55:549–564

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g., a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.