https://doi.org/10.1016/j.eswa.2020.113216

Yin-Yang Firefly Algorithm Based on Dimensionally Cauchy Mutation

Wen-chuan Wang
School of Water conservancy, North China University of Water Resources and Electric Power, Zhengzhou, 450046, P.R. China
Corresponding author, E-mail: wangwen1621@163.com; wangwenchuan@ncwu.edu.cn

Lei Xu
School of Water conservancy, North China University of Water Resources and Electric Power, Zhengzhou, 450046, P.R. China
E-mail: xulei234@foxmail.com

Kwok-wing Chau
Department of Civil and Environmental Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, P.R. China
E-mail: cekwchau@polyu.edu.hk

Dong-mei Xu
School of Water conservancy, North China University of Water Resources and Electric Power, Zhengzhou, 450046, P.R. China
E-mail: xudongmei@ncwu.edu.cn

Abstract

Firefly algorithm (FA) is a classical and efficient swarm intelligence optimization method and has a natural capability to address multimodal optimization. However, it suffers from premature convergence and low stability in the solution quality. In this paper, a Yin-Yang firefly algorithm (YYFA) based on dimensionally Cauchy mutation is proposed for performance improvement of FA. An initial position of fireflies is specified by the good nodes set (GNS) strategy to ensure the spatial representativeness of the firefly population. A designed random attraction model is then used in the proposed work to reduce the time complexity of the algorithm. Besides, a key self-learning procedure on the brightest firefly is undertaken to strike a balance between exploration and exploitation. The performance of the proposed algorithm is verified by a set of CEC 2013 benchmark functions used for the single objective real parameter algorithm competition. Experimental results are compared with those of other the state-of-the-art variants of FA. Nonparametric statistical tests on the results demonstrate that YYFA provides highly competitive performance in terms of the tested algorithms. In addition, the application ~~of a~~ in constrained engineering optimization problems shows the practicability of YYFA algorithm.

Keywords

Yin-Yang firefly algorithm; Cauchy mutation; GNS strategy; Random attraction model; CEC 2013 benchmark functions; Engineering optimization problems

1. Introduction

Firefly algorithm (FA) is a swarm intelligence algorithm based on flashing patterns and behavior of fireflies (Yang, 2014). It has advantages of simple structure and easy operation, and has been widely used in structural optimization (Chou & Ngo, 2017; Kaveh, Mahdipour Moghanni, & Javadi, 2019), engineering prediction (Danandeh Mehr, Nourani, Karimi Khosrowshahi, & Ghorbani, 2019; Tao, et al., 2018), resource allocation (Garousi-Nejad, Bozorg-Haddad, Loáiciga Hugo, & Mariño Miguel, 2016; H. Wang, et al., 2018) and other fields (Mosavvar & Ghaffari, 2019; Rajinikanth & Couceiro, 2015). However, it has a defect of low convergence accuracy in the process. Therefore, scholars have improved the firefly algorithm from several perspectives. The list of main variants of FA with their characteristics is shown in **Table 1**. It can be summarized that FA could be improved in seven aspects: adaptive parameters, novel move mode, novel attraction mode, elitism strategy, multi-groups, hybrid algorithm and interdisciplinary application. The following is a discussion on the characteristics of these seven aspects.

Table 1

• The strategy of adaptive parameters has been one of the most popular techniques utilized in FA. Wang et al. (2017) found that the attractiveness had kept unchangeable at $\beta 0$ (which referred to the initial value of attractiveness) since an extremely early stage during the search process in standard FA. Then a simple dynamic strategy to adjust the attractiveness coefficient has been applied to tackle this problem. Otherwise, chaotic maps also played an important role in adjusting parameters.

• The improvement in move modes tried to enhance the search capability and reduce the possibility of population oscillation. This strategy included different approaches from different perspectives. Tian et al. adopted a time-varying inertia weight method for the current location of fireflies (Tian, et al., 2012). The simulation results indicated that IWFA outperformed FA and PSO. Uniform distribution, Gaussian distribution and Lévy flight were introduced into the randomization term of movement and had shown promising capabilities.

• The strategy of novel attraction mode aimed to reduce the computational complexity of FA. Specific methods have been employed to choose one or more brighter fireflies to move. The time saved can be used to implement other improvement strategies.

• The elitism strategy helped make the brightest firefly in the swarm or other fireflies brighter. RaFA utilized the Cauchy jump to update the brightest firefly for accelerating convergence; ODFA adopted an opposition-based learning method and dimensional-based approach to ensure the superiority of the population before the movement process (Verma, et al., 2016); OLFA used an orthogonal learning technique to generate a promising learning exemplar for every firefly (Tomas, et al., 2019).

- Dividing all fireflies into groups to implement different strategies has also been an effective way to improve the performance of FA. This method greatly enriched the diversity of the population. As a typical example, the firefly colony in IMGFA was divided into several subgroups with different model parameters (Tong, et al., 2017). Each subgroup carried out its own internal independent operation, and then the brightest firefly of each subgroup exchanged information. From this point of view, this method reduced the operability of the algorithm to a certain extent.

- The ability of a single optimization algorithm was often flawed. FA did not perform well in searching for global optimum at a later stage of the iteration process. Hybrid algorithm has been an effective method to combine FA with other robust techniques. Namely, a tool with a strong local search ability, such as FA-PS, HFADE, HS/FA and CEFA, was embedded into a weak link of FA (Guo, et al., 2013; Li, et al., 2019; Sarbazfard & Jafarian, 2016; Wahid & Ghazali, 2019). In particular, FAPSO was different from hybrid algorithms. The main idea in FAPSO was multi-groups, namely two sub-populations selecting FA and PSO as their basic algorithm, to carry out the optimization process respectively (Xia, et al., 2018).

- Interdisciplinary application denoted that an inspiration from other disciplines could help improve FA. FAtidal algorithm applied the Tidal Force formula (Yelghi & Köse, 2018), which described the effect of a massive body that gravitationally affected another massive body, to strengthen the exploitation function of FA. QFA algorithm adopted quaternion to represent the individuals in FA. However, QFA did not show any particular superiority according to their experimental results. In general, this strategy lost the simplicity of the FA.

In general, the standard FA has a simple structure and strong operability. Its optimization ability depends on the brightest firefly in the swarm, which has a weak function in exploration if the brightest firefly gets trapped in the local optimum. Otherwise, FA does not perform deep information mining for the brightest firefly during the iteration. As such, we try to reduce the number of times for movements and allocate computing resources to perform actions on the brightest firefly for attaining a good balance between the functions of exploration and exploitation. Therefore, an effective method named *Cauchy mutation* is applied to modify the FA algorithm, by which Yin-Yang firefly algorithm (YYFA) is proposed. The main procedure of YYFA is stated as follows. A new random attraction model is firstly designed to replace the full attraction model in the original FA algorithm to reduce wastage of computing resources. Secondly, a self-learning strategy based on the elitism strategy with Cauchy mutation is utilized to strengthen the exploration and exploitation functions. Furthermore, a good nodes set (GNS) strategy is used to initialize the firefly population in order to improve the spatial representativeness of the population.

The structure of the paper is organized as follows. In the next section, the basic theory of FA, Cauchy mutation and GNS strategy are discussed. The proposed YYFA algorithm is described and discussed in Section 3. Section 4 shows the behavior of the new approach and nonparametric statistical tests are employed on experimental results to analyze the performance of the proposed algorithm. In Section 5, four well-known engineering constrained optimization problems and a storm intensity model problem are utilized to further verify the performance of the proposed YYFA algorithm. Finally, the work is summarized in Section 6.

2. Preliminary

2.1 Firefly algorithm

Let D be the dimension of the search space. The location of each firefly in the search space represents a feasible solution, and its brightness represents the fitness of the optimization problem. Then, according to the fact that fireflies move in turn to brighter fireflies than themselves, the location update formula of firefly i attracted by a brighter firefly j is defined as:

$$x_{id}(t+1) = x_{id}(t) + \beta(x_{jd}(t) - x_{id}(t)) + \alpha\left(t\right)\varepsilon_i \tag{1}$$

where x_{id} and x_{jd} are the d-dimensional positions of the firefly i and j, respectively. β is the attractiveness, α represents the step factor, t indicates the iteration number and ε obeys uniform distribution in the range of [-0.5, 0.5].

α in the standard firefly algorithm is defined by:

$$\alpha\left(t\right) = \alpha_0 \theta^{\,t} \tag{2}$$

where α_0 is the initial step factor of the algorithm, which is taken as 1; θ is the cooling coefficient and the range of values is [0.95, 0.99] (Yang, 2014).

The brightness and attractiveness of a firefly can be computed by:

$$I = I_0 \exp(-\gamma r_{ij}^2) \tag{3}$$

$$\beta = \beta_0 \exp(-\gamma r_{ij}^2) \tag{4}$$

where β_0, I_0 are the attractiveness and brightness, respectively, at the location of the firefly itself, namely r=0, and r is the distance between two fireflies computed by:

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2} \tag{5}$$

If we consider minimization problems, the framework of the standard FA is shown in **Figure 1**.

Figure 1

2.2 Cauchy mutation

Cauchy mutation is an efficient technique for improving optimization algorithms (Hu, Wu, Wang, & Xie, 2009; Ali & Pant, 2011; Sapre & Mini, 2019). The theoretical basis of Cauchy mutation is Cauchy probability density function, which is defined by Equation (6). Curves of Cauchy density function and standard normal distribution density function are presented in **Figure 2**. It should be noted that the red curve is the standard Cauchy density curve. From the figure, the Cauchy distribution curves have long fat tails compared with the standard normal distribution, which can help the firefly jump out from the local optimum. Wang et al. (2016) conducted a Cauchy mutation in the firefly algorithm by Equation (7):

$$f(x) = \frac{1}{\pi}\left[\frac{a}{(x-x_0)^2 + a^2}\right] \tag{6}$$

$$X_{best}^{d\ *} = X_{best}^{d} + cauchy \tag{7}$$

where X_{best}^{d} denotes the d_{th} dimension position of the best firefly found so far and *Cauchy* is a random number generated by the standard Cauchy distribution.

However, it can be seen from **Figure 2** that the standard Cauchy distribution falls within the interval of [-5,5] with a high probability. When faced with the optimization problem of large search range, Cauchy mutation is not adaptive to perform as the second term on the right side of Equation (7). Therefore, the equation needs to be redesigned to meet the universality for more optimization problems.

Figure 2

2.3 GNS strategy

In the swarm intelligence algorithm, we are eager to obtain better information from the initial firefly population, which means that the initial fireflies should be able to reflect the spatial characteristics in the search space. In other words, only when a population of fireflies which can best reflect the spatial characteristics in the search space is taken as the initial population, can the optimization quality be improved. Based on this idea, we attempt to initialize the position of fireflies by the good nodes set (GNS) strategy (Xiao, Cai, & Wang, 2007). The deviation of points generated by using the good nodes set strategy was much smaller than those of randomly selected points in theory (Hua & Wang, 1978). For comparison, we construct two point sets as shown in **Figure 3**. The left one is a set containing 100 two-dimensional good points in unit space. In the right one, 100 points are selected in two-dimensional unit space by a random method. The distribution of good point sets is obviously more even than that of random points. For the firefly algorithm, this method can avoid the generation of invalid fireflies and accelerate the convergence speed.

Figure 3

3 Yin-Yang firefly algorithm

3.1 Designed attraction model

An evolutionary updating of a swarm in the standard firefly algorithm is accomplished by using a full attraction model, namely, each firefly moves in turn to a brighter one in each iteration.

Let *N* be the number of fireflies in the swarm, so the maximum number of moves needed in each iteration is $M_f = N*(N-1)/2$. This will lead to wastage of computing resources and oscillation when fireflies approach the global optimum. In order to save computing resources, Wang et al. (2016) proposed a random attraction model, that is, the current firefly randomly selected a firefly from the swarm and judged its brightness to choose whether to move or not. Inspired by that study, this study adopts a new random attraction model to replace the full attraction model to meet the exploration function of Yin-Yang firefly algorithm.

In the random attraction model of Yin-Yang firefly algorithm, the first step is to ensure that individual brightness of the input swarm ranks from strong to weak. In the moving process of fireflies, we hope that weaker fireflies will become brighter when they move to brighter ones. In the proposed model, we hold that fireflies can maintain this trend without extra measures to avoid possible influence of weaker brightness fireflies. The main step of the random attraction model is described in the following Algorithm of Firefly Moving.

Figure 4

As shown in **Figure 4**, the proposed model starts with the second firefly, each firefly randomly selects one from the fireflies prior to move. Next come the third and fourth fireflies, and so on to the *N*th firefly to ensure the diversity of the swarm. Thus, the total number of moves needed in each iteration is $M_r = N-1$. With the increase of number of fireflies and number of iterations, this new attraction model consumes less computational resources than the full attraction model, and more computational resources can be used for the next Yin-Yang firefly self-learning strategy.

3.2 Yin-Yang firefly self-learning strategy

The theory of Yin-Yang in ancient China is the crystallization of wisdom of laboring people. It emphasizes the law of "mutual survival of negative and positive" and "balance between Yin and

Yang" in the world. The algorithm also focuses on seeking a balance between the two opposite functions of exploration and exploitation to attain better solutions. Therefore, the proposed algorithm adopts a Yin-Yang firefly self-learning strategy to explore the search space as well as to undertake high-level data mining for the optimal firefly.

After a position update of the firefly swarm, the Yin-Yang firefly algorithm selects the firefly X_p with the best fitness as the "Yang firefly" and gives it a certain time for self-learning. Then a new firefly X_o is created randomly in the search space as a "Yin firefly". In a single learning process to address the shortcoming of Equation (7), the position of X_o is updated and modified in single dimension according to Equation (8).

$$X_o^d = X_p^d + cauchy \cdot \left(X_{r1}^d - X_{r2}^d \right) \quad (8)$$

where X_o^d, X_p^d denote the d^{th} dimension positions of the Yin and Yang fireflies, respectively; *Cauchy* represents a stochastic number generated by the standard Cauchy distribution function; and X_{r1}^d, X_{r2}^d are the d-dimensional positions of two fireflies randomly selected from the swarm.

From the above equation, a multiplicative term related to the size of global domain is added to the Cauchy mutation item. Therefore, in the early stage of algorithm optimization, the population is evenly distributed. The brightest fireflies can adaptively learn based on the size of the search space to avoid missing local space due to the limitation of the Cauchy distribution. After updating the position, the fitness of X_o will be evaluated and compared with that of X_p. If the fitness of X_o is worse, it continues to update X_o in the next dimension. Once the exploration gets successful, namely the fitness of firefly X_o is better than that of X_p, the position and fitness of X_o are assigned to X_p to realize the balance between Yin and Yang, at which time both fireflies are the current optimal fireflies. The optimal firefly will use the remaining learning times to undertake deep data mining to meet the exploitation function of the algorithm.

3.3 Framework of the proposed YYFA

The step factor α and attractiveness β in the proposed approach are updated by Equation (9) (H. Wang, Zhou, et al., 2017) and Equation (10) (J. I. Fister, Xin-She, Iztok, & Janez, 2012), respectively.

$$\alpha\left(t+1\right)=\alpha\left(t\right)\cdot\left(1-\frac{t}{T}\right) \tag{9}$$

$$\beta=\beta_{\min}+\left(\beta_0-\beta_{\min}\right)e^{-\gamma r_{ij}^2} \tag{10}$$

where $\beta_{\min}$ is the minimum value of attractiveness; T is the maximum number of generations; and other parameters have the same meanings as before.

Combining the GNS strategy, specially-designed attraction model and Yin-Yang firefly self-learning strategies, the pseudo code of our proposed YYFA algorithm is shown in **Figure 5**.

Figure 5

3.4 Analysis of YYFA

3.4.1 Computational complexity

Let D be the dimension of the objective function, N be the swarm size, T be the maximum number of iterations, L be the self-learning time for Yin and Yang fireflies, and F be the computational time for evaluating the objective function. Then the maximum time consumptions TC of YYFA algorithm and FA algorithm are respectively:

$$TC_{YYFA}=\left(N+\left(D+\frac{N\text{-}1}{L}\right)*T\right)*F+\left(D+\frac{N-1}{L}\right)*T \tag{11}$$

$$TC_{FA}=\left(N+\frac{N(N-1)}{2}*T\right)*F+\frac{N(N-1)}{2}*T \tag{12}$$

The time consumption of firefly algorithm is mainly composed of two parts: the first part is the time consumption for evaluating the objective function, and the second part is the time consumption for the moves. As can be seen from Equation (11), since L is generally set to be much larger than N, TC_{YYFA} can be approximated as:

$$TC_{YYFA}=(N+D*T)*F*T+D*T \tag{13}$$

By utilizing the O notation to analyze the computational complexity, the computational complexity of YYFA is $O(D)$ and that of FA is $O(N^2)$. In general, D is in the same order of magnitude as N. Thus, YYFA algorithm has a lower computational complexity.

3.4.2 Comments on parameters

In YYFA, we adopt the parameter setting of $\alpha(0)=0.2$, $\beta_{\min}=0.2$, $\beta_0=1$ and $\gamma=1$ for attractions and moves. In addition, the parameters required from the user are the population size N and the number of self-learning times L for brightest firefly. From **Subsection 3.4.1**, the time complexity of YYFA is directly proportional to the dimension of problem rather than the number of fireflies. Thus, we can initialize the population by more fireflies to make the most of GNS strategy. Too many fireflies, however, would reduce the distance between individuals and lead to fluctuations. L should be defined based on the problem size and number of iterations and thus controls the frequency of population movements. A large value of L will help improve in finding a better position for the current brightest firefly and local search, but easily get a slow convergence rate and lose the effectiveness of other fireflies. On the other hand, a low value of L will accelerate the algorithm but can be stuck in premature convergence.

4. Simulations and experiments

4.1 Algorithm behavior

In this section, eight two-dimensional test functions are simulated to denomstrate behaviors of the proposed YYFA algorithm during the optimization process. Details of the above functions are presented in **Table 2** and they are all minimization problems. We use five fireflies to test each function in 5000 iterations coupled with 50 self-learning times, and the search results are shown in the column of '*Search result*' in **Table 2**. **Figure 6** shows the two-dimensional test function and paths of firefly population on the contour plots.

As can be seen from **Table 2**, YYFA algorithm has promising results on these eight test functions. Five fireflies can accurately find the global best in four of them. The results of Levy N. 13 function and Rosenbrock function are very close to the global best. The errors of the remaining two functions can also be controlled within 0.001. The followings are some observations via inspecting behaviors of fireflies in **Figure 6**:

(i) The initialization by GNS strategy renders fireflies evenly distributed, so that only 5 fireflies can attain reliable results to save computing resources;

(ii) The population can be guided and moved to the global optimum by the self-learning process. Taking the Levy N. 13 function as an example, its global optimum is located near the center of the search space, and 5 fireflies are initially distributed around the periphery of the search space. After the first time of Yin-Yang firefly self-learning process, the fireflies quickly gathered from different directions to the optimum.

(iii) YYFA has the capability of local search. Bukin function has many local bests around the global optimum. From **Figure 6 (a)**, it can be seen that when the firefly population is near the global optimum, the population starts to mine effective information in a surrounding manner.

Table 2

Figure 6

4.2 Benchmark functions and simulation environment

The suite of 28 benchmark functions used for the Single Objective Real Parameter Algorithm competition that was held in the Congress on Evolutionary Computation 2013 (CEC 2013) is utilized to test the proposed YYFA algorithm. The benchmarks can be classified into three categories: unimodal functions (f_1–f_5), basic multimodal functions (f_6–f_{20}) and composition functions (f_{21}–f_{28}). The function names along with their global optima are provided in **Table 3**. For more details on these, please refer to Liang, et al. (2013).

Table 3

The variable bounds for all dimensions of the functions are specified as [-100, 100] and the corresponding global optimum value does not change with dimensions. The competition requires that the algorithm be tested for three dimension-settings (D=10, 30 and 50) along with the corresponding maximum number of functional evaluations ($D*10^4$). To maximize the ability of the algorithm, we use the corresponding maximum number of iterations ($D*10^4$) as a stopping criterion. With a fixed number of iterations, the number of function evaluations for each optimization of FA could be different. Thus, the number of function evaluations consumed by algorithms in each test will be recorded to help further analysis.

Additionally, all the experiments on a single function will run 51 times independently to eliminate the impact of randomness. All results are recorded in terms of error between the global optimum and value obtained by the algorithm. The terms 'Mean', 'Std. dev.' and 'Num. of Eval.' refer to the mean, standard deviation of the error and mean number of function evaluations obtained over 51 runs. All experiments are run on a Windows 10 64-bit computer with an Intel i7 (3.4GHz) processor and 8 GB RAM, and are implemented under MATLAB R2018a environment.

4.3 Numerical experiments and results discussion

In order to test the performance of YYFA algorithm, FA and three state-of-the-art FA variants are selected for comparison. They are ApFA (H. Wang, Zhou, et al., 2017), RaFA (H. Wang, et al., 2016) and OBLFA (Yu, et al., 2015a). The comparative study in this section is based on the 28 benchmarks in CEC 2013 competition.

Parameter settings are vital to the performance of the algorithm. The GNS strategy in YYFA requires a large population number N to guarantee the performance of the algorithm. Considering the fairness of the test and the characteristics of other contestants, however, the population size N is set to be 20, 30 and 40 for the three dimension-settings as the complexity of the problem increases. Thus, the self-learning times L in YYFA is set to a large value, which are 800 for 10D, 30D cases and 625 for 50D case. This will slow down the convergence speed of the YYFA and consume more computing resources to some extent. The settings of other parameters for each algorithm adopt the values recommended in the original literature, which are presented in **Table 4**. Since RaFA and OBLFA do not provide ideal parameter updating equations for α and β, we adopt the same equations as for YYFA.

Table 2.

The performance of test algorithms on the benchmarks at dimensions 10, 30 and 50 are provided in **Table 5**, **6** and **7** respectively. It can be clearly seen in **Table 5** that YYFA outperforms RaFA, OBLFA and FA for most test functions. But OBLFA and FA can achieve slightly better mean error than YYFA on function f_{21} and f_{16}, respectively. Besides, YYFA gets better results in terms of mean error and standard deviation on 13 functions compared with ApFA. As for the mean number of function evaluations over 51 runs, OBLFA consumes the most resource to evaluate in general while YYFA needs slightly more function evaluations than ApFA. In the 30D case from **Table 6**, YYFA still maintains its advantage in convergence accuracy over RaFA, OBLFA and FA but ranks last on f_{16}. In addition, YYFA has only 11 functions tested with better results in comparison with ApFA and consumes more computational resources to get a better fitness such as function f_4 and f_{14}. This also validates our thinking in **subsection 3.4.2** about setting parameters, which refers to that the parameter L of 625 is relatively large to slow the convergence speed. The ability of algorithm to search the global optimum would deteriorate along with increase in the problem dimension, but YYFA is still able to determine such values on function f_1, f_4, f_5, f_{11} and f_{14} in 50D case. In this case, YYFA obtains better mean accuracy than ApFA on 12 functions but get stuck in more function evaluation times.

To quantitatively analyze the differences between the test algorithms, we conduct pairwise comparisons based on the Wilcoxon signed rank test (Derrac, García, Molina, & Herrera, 2011). This test analyzes the significance of the difference between two algorithms by checking whether the two sets of samples come from different population distributions. In this study, the mean errors and its corresponding standard deviations are taken as the test data. The results are presented in **Table 8**, where R^+ is the sum of ranks for the problems in which YYFA outperforms the competing algorithm and p-value associated with min (R^+, R^-). As this table shows, the null hypothesis which holds that the two algorithms are the same, is rejected considering a significance value of α=0.05 for all comparisons with RaFA, OBLFA and FA over three dimension settings. Combined with the values of R^+, we can hold that YYFA has a superior performance over them. Furthermore, p-values from ApFA all exceed 0.1, which means the hypothesis is accepted and YYFA has the same performance as ApFA statistically.

The convergence curves for some selected functions on all dimension cases are presented in **Figure 7**, **8** and **9**. The followings are some observations as inferred from the curves:

- Compared with other test algorithms, YYFA has the slowest convergence speed, which is consistent with our comments on parameters discussed in **Subsection 3.4.2**.
- The curves of YYFA on f_1 in 10D case and f_5 suddenly fall almost vertically in the process, which shows its ability of escape-local-optimum.
- In 10D case, although the convergence of YYFA at early stage is slower on f_4, f_6, f_{14} and f_{17}, the algorithm provides lower errors at the end.
- In 50D case, YYFA show its great performance on composition functions f_{22}, f_{26} and f_{28}, which indicates that YYFA is an effective approach to address complicated problems.

Table 3

Table 6

Table 7

Table 8

Figure 7

Figure 8

Figure 9

4.4 Parameter sensitivity of YYFA

In **Section 4.3**, we test the proposed YYFA algorithm and other FA variants. The results verify the effectiveness of the modified Equation (8) based on RaFA and proves its advanced status. However, the shortcoming in convergence speed of YYFA is also a key problem that cannot be ignored. Thus, 10 different combinations of the two user-defined parameters N and L are employed to provide insights into effects of these parameters compared with the base setting in **Section 4.3**. We conduct the experiments based on 6 selected functions in 30D case including f_2, f_6, f_{15}, f_{20}, f_{21} and f_{28}, which ensure the integrity of function categories (f_2 is a unimodal function, f_6, f_{15}, f_{20} are multimodal functions and f_{21}, f_{28} belong to composition functions). The details of combinations and the results over 51 independent runs are presented in **Table 9**. The convergence curves for different combinations on each function are given in **Figure 10**. The followings are observations from the results and curves on three different function categories.

Unimodal function f_2: Comb. 4 with N=100 and L=250 reduces the mean error by almost three-quarters but consumes less computing resources according to the base case. To compare with ApFA, it is meaningful for YYFA with Comb. 4 to reduce the error by about an order of magnitude with more function evaluations. From the curves, we can observe that combinations with a low value of L (Comb. 2 and 10) converge fastest but miss a better result while combinations with a high value of L (Comb. 1 and 9) have a slowest speed. Besides, the parameter N has not much impact on results under the same L.

Multimodal functions f_6,f_{15},f_{20}: Function f_6 has about a similar situation as f_2 with L dominating. Comb. 4 with N=100 and L=250 attains the best fitness with less times to evaluate. The results on f_{15} among 10 combinations are close. The best one is still worse compared with ApFA, which verifies the No Free Lunch theorem (Wolpert & Macready, 1997) that YYFA fails to search on f_{15}. Comb. 8 with N=500 and L=500 makes great difference on f_{20}. When the optimization results of other parameter combinations (except Comb. 2) are limited to about 15, the mean error obtained by Comb. 8 can fall below 13. It can be inferred that YYFA algorithm prefers a large value of N instead of ordinary value below 100.

Composition functions f_{21}, f_{28}: From the curves of f_{21}, we can observe that although Comb. 9 with N=250 and L=2000 converge slowest, it helps f_{21} get the smallest mean error, which is superior to ApFA. This also proves the former parameter discussion that a large value of L will help local search. Several groups of parameters achieve more reliable results on f_{28}, and the group with larger L accounts for the majority among them.

To summarize, YYFA algorithm is able to attain a reliable result with moderate number of function evaluations. The ideal value of parameter N for the optimization problem should be large enough firstly. Besides, parameter L is set according to the problem's dimension, the prefer L is supposed to be moderate. Parameter tuning procedure (Eiben & Smit, 2011) could also be employed.

Table 9

Figure 10

5. Performance in practical optimization problems

5.1 Constrained engineering optimization problems

This section is devoted to the performance evaluation of the proposed YYFA algorithm on four well-known constrained engineering optimization problems, which are problems of *pressure vessel design* (*PVD*), *tension/compression spring* (*TCS*), *welded beam design* (*WBD*) and *speed reducer design* (*SRD*). Details of constraints and ranges for these problems can be referred to Baykasoğlu & Ozsoydan (2015). All problems belong to minimization questions while satisfying the constraints. To handle the constraints, a basic penalty method (considering a penalty factor of 10^{30}) is employed when the problem encounters a constraint violation. Fifty independent tests are run for each problem and the best solution are recorded and compared with ApFA in **Table 10**.

As it can be seen from the table, the results are straightforward since YYFA has competitive fitness values in addressing the four problems. It can be observed that YYFA consumes fewer function evaluations and gets better fitness values than ApFA. From the above, YYFA is suggested as a helpful solver for constrained single-objective optimization problems.

Table 10

5.2 Parameters optimization in rainstorm intensity model

The joint effects of global climate change and urbanization have a significant impact on urban flood control safety. To alleviate the problem of flood, we must strengthen the construction of urban drainage and waterlogging prevention infrastructure. The important premise is to scientifically determine a reasonable equation for urban rainstorm intensity. Equation (14) is often used to compute the intensity of rainstorm in a single recurrence period.

$$i = \frac{M}{(t+n)^b} \tag{14}$$

where i denotes the rainstorm intensity (mm/s); t indicates the duration of rainfall (min); M, n and b are some parameters.

As the equation is an overdetermined nonlinear equation, the parameter optimization problem of the equation is actually a nonlinear optimization problem. In this work, YYFA and FA are used respectively to optimize the parameters for the real rainstorm data. The adopted fitness function is:

$$\min Q = \sum_{k=1}^{m} \left(\frac{M}{(t_k+n)^b} - i_k \right)^2 \tag{15}$$

where Q denotes the residual sum of squares, k is the serial number of the specific rainfall duration and i_k represents the real rainstorm intensity.

The real data containing the relationship between the intensity and duration of rainstorm in three different recurrence periods in Zhengzhou City are chosen from Tang, Zhang, Wang, & Liu (2019) as shown in **Table 11**. Besides, the search range for model parameters is set as M=[0,100], n=[0,100] and b=[0,2]. Both two algorithms run for 30 independent times and the best parameter estimates are recorded in **Table 12**.

Table 11

Table 12

It can be observed that YYFA algorithm has a better performance on the rainstorm intensity model than FA, which proves the practicability of YYFA.

6. Conclusions

An improved firefly algorithm based on the Yin Yang philosophy, named Yin-Yang firefly algorithm, for single-objective optimization problems is proposed to strike a balance between exploitation and exploration by the modified dimensional Cauchy mutation. The framework of YYFA is presented in details with analysis of its time complexity and sensitivity of user-defined parameters. The proposed algorithm is compared with the state-of-the-art FA variants based on CEC 2013 benchmark functions and it is verified that YYFA has a competitive performance. Besides, we make some suggestions on parameter selection. Its applications in four popular constrained engineering optimization problems demonstrate its advancement. Based on our analysis, YYFA has several particular features as listed below:

- YYFA has a simple structure and strong programmability with only one equation added for Cauchy mutation on the brightest firefly. The design of Cauchy mutation on each dimension results in a decrease in time complexity, which leads to transform in large population size for GNS strategy.
- To the best of our knowledge, this work is the first one to employ the technique of GNS in FA, which helps enhance the algorithm performance through large population size.
- Different combinations of user-defined parameters gives more chances to attain reliable solutions, which is proven by results on four popular constrained engineering optimization problems.

The paper proves that YYFA has a good optimization potential. The follow-up work is to employ techniques such as orthogonal experiment design to conduct a more rigorous study on two user-defined parameters and apply YYFA to dynamic optimization problems as well as more practical optimization problems.

Acknowledgements

The authors are grateful to the supports of Henan province university scientific and technological innovation team (No: 18IRTSTHN009), National Natural Science Foundation of

China (No: 51509088), and Henan Key Laboratory of Water Environment Simulation and Treatment (No: 2017016).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

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

References

Ali, M., & Pant, M. (2011). Improving the performance of differential evolution algorithm using Cauchy mutation. *Soft Computing, 15*, 991-1007.

Baykasoğlu, A., & Ozsoydan, F. B. (2014). An improved firefly algorithm for solving dynamic multidimensional knapsack problems. *Expert Systems with Applications, 41*, 3712-3725.

Baykasoğlu, A., & Ozsoydan, F. B. (2015). Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Applied Soft Computing, 36*, 152-164.

Cheung, N. J., Xue-Ming, D., & Hong-Bin, S. (2014). Adaptive Firefly Algorithm: Parameter Analysis and its Application. *PLoS ONE, 9*, 1-12.

Chou, J.-S., & Ngo, N.-T. (2017). Modified firefly algorithm for multidimensional optimization in structural design problems. *Structural and Multidisciplinary Optimization, 55*, 2013-2028.

Danandeh Mehr, A., Nourani, V., Karimi Khosrowshahi, V., & Ghorbani, M. A. (2019). A hybrid support vector regression–firefly model for monthly rainfall forecasting. *International Journal of Environmental Science and Technology, 16*, 335-346.

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 and Evolutionary Computation, 1, 3-18.

Eiben, A. E., & Smit, S. K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation, 1*, 19-31.

Farahani, S. M., Abshouri, A. A., Nasiri, B., & Meybodi, M. R. (2011). An Improved Firefly Algorithm with Directed Movement. In 2011 4th IEEE International Conference on Computer Science and Information Technology(ICCSIT 2011). Chengdu China.

Fister, I., Yang, X.-S., Brest, J., & Fister, I. (2013). Modified firefly algorithm using quaternion representation. *Expert Systems with Applications, 40*, 7220-7230.

Fister, J. I., Xin-She, Y., Iztok, F., & Janez, B. (2012). Memetic firefly algorithm for combinatorial optimization. *Mathematics*, arXiv:1204.5165.

Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation, 18*, 89-98.

Garousi-Nejad, I., Bozorg-Haddad, O., Loáiciga Hugo, A., & Mariño Miguel, A. (2016). Application of the Firefly Algorithm to Optimal Operation of Reservoirs with the Purpose of Irrigation Supply and Hydropower Production. *Journal of Irrigation and Drainage Engineering, 142*, 04016041.

Guo, L., Wang, G.-G., Wang, H., & Wang, D. (2013). An Effective Hybrid Firefly Algorithm with Harmony Search for Global Numerical Optimization. *The Scientific World Journal $V 2013*, 1-9.

Hassanzadeh, T., & Kanan, H. R. (2014). FUZZY FA: A MODIFIED FIREFLY ALGORITHM. *Applied Artificial Intelligence, 28*, 47-65.

Hu, C., Wu, X., Wang, Y., & Xie, F. (2009). Multi-swarm Particle Swarm Optimizer with Cauchy Mutation for Dynamic Optimization Problems. In Z. Cai, Z. Li, Z. Kang & Y. Liu (Eds.), *Advances in Computation and Intelligence* (pp. 443-453). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hua, L.-g., & Wang, Y. (1978). *The Application of Number Theory in Approximate Analysis*: Science Press, Beijing.

Kaveh, A., Mahdipour Moghanni, R., & Javadi, S. M. (2019). Optimum design of large steel skeletal structures using chaotic firefly optimization algorithm based on the Gaussian map. *Structural and Multidisciplinary Optimization, 60*, 879-894.

Li, G., Liu, P., Le, C., & Zhou, B. (2019). A Novel Hybrid Meta-Heuristic Algorithm Based on the Cross-Entropy Method and Firefly Algorithm for Global Optimization. *Entropy, 21*, 494.

Liang, J., Qu, B., Suganthan, P., & Hernández-Díaz, A. (2013). Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization. *Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China*.

Lv, L., & Zhao, J. (2018). The Firefly Algorithm with Gaussian Disturbance and Local Search. *Journal of Signal Processing Systems, 90*, 1123-1131.

Mosavvar, I., & Ghaffari, A. (2019). Data Aggregation in Wireless Sensor Networks Using Firefly Algorithm. *Wireless Personal Communications, 104*, 307-324.

Pan, X., Xue, L., & Li, R. (2019). A new and efficient firefly algorithm for numerical optimization problems. *Neural Computing and Applications, 31*, 1445-1453.

Rajinikanth, V., & Couceiro, M. S. (2015). RGB Histogram Based Color Image Segmentation Using Firefly Algorithm. *Procedia Computer Science, 46*, 1449-1457.

Sapre, S., & Mini, S. (2019). Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization. *Soft Computing, 23*, 6023-

6041.

Sarbazfard, S., & Jafarian, A. (2016). A Hybrid Algorithm Based on Firefly Algorithm and Differential Evolution for Global Optimization. *International Journal of Advanced Computer Science and Applications, 7*.

Tang, Y., Zhang, Y., Wang, H., & Liu, Y. (2019). Algorithm of Rainstorm Intensity Formula Optimization Based on NicheAGA-CGA. *Journal of Beijing University of Technology, 45*, 292-298.

Tao, H., Diop, L., Bodian, A., Djaman, K., Ndiaye, P. M., & Yaseen, Z. M. (2018). Reference evapotranspiration prediction using hybridized fuzzy model with firefly algorithm: Regional case study in Burkina Faso. *Agricultural Water Management, 208*, 140-151.

Tian, Y., Gao, W., & Yan, S. (2012). An Improved Inertia Weight Firefly Optimization Algorithm and Application. In *2012 International Conference on Control Engineering and Communication Technology* (pp. 64-68).

Tomas, K., Michal, P., Adam, V., & Roman, S. (2019). Firefly Algorithm Enhanced by Orthogonal Learning. In R. Silhavy (Ed.), *Artificial Intelligence and Algorithms in Intelligent Systems* (pp. 477-488). Cham: Springer International Publishing.

Tong, N., Fu, Q., Zhong, C., & Wang, P. (2017). A multi-group firefly algorithm for numerical optimization. *Journal of Physics: Conference Series, 887*, 012060.

Verma, O. P., Aggarwal, D., & Patodi, T. (2016). Opposition and dimensional based modified firefly algorithm. *Expert Systems with Applications, 44*, 168-176.

Wahid, F., & Ghazali, R. (2019). Hybrid of firefly algorithm and pattern search for solving optimization problems. *Evolutionary Intelligence, 12*, 1-10.

Wang, B., Li, D.-X., Jiang, J.-P., & Liao, Y.-H. (2016). A modified firefly algorithm based on light intensity difference. *Journal of Combinatorial Optimization, 31*, 1045-1060.

Wang, C.-F., & Song, W.-X. (2019). A novel firefly algorithm based on gender difference and its convergence. *Applied Soft Computing, 80*, 107-124.

Wang, H., Wang, W., Cui, Z., Zhou, X., Zhao, J., & Li, Y. (2018). A new dynamic firefly algorithm for demand estimation of water resources. *Information Sciences, 438*, 95-106.

Wang, H., Wang, W., Sun, H., & Shahryar, R. (2016). Firefly algorithm with random attraction. *International Journal of Bio-Inspired Computation, 8*, 33-41.

Wang, H., Wang, W., Zhou, X., Sun, H., Zhao, J., Yu, X., & Cui, Z. (2017). Firefly algorithm with neighborhood attraction. *Information Sciences, 382-383*, 374-387.

Wang, H., Zhou, X., Sun, H., Yu, X., Zhao, J., Zhang, H., & Cui, L. (2017). Firefly algorithm with adaptive control parameters. *Soft Computing, 21*, 5091-5102.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1, 67-82.

Xia, X., Gui, L., He, G., Xie, C., Wei, B., Xing, Y., Wu, R., & Tang, Y. (2018). A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm. *Journal of Computational Science, 26*, 488-500.

Xiao, C., Cai, Z., & Wang, Y. (2007). A good nodes set evolution strategy for constrained optimization. In *2007 IEEE Congress on Evolutionary Computation* (pp. 943-950).

Yang, X.-S. (2010). Firefly Algorithm, Lévy Flights and Global Optimization. In M. Bramer, R. Ellis & M. Petridis (Eds.), *Research and Development in Intelligent Systems XXVI* (pp. 209-218). London: Springer London.

Yang, X.-S. (2014). Chapter 8 - Firefly Algorithms. In X.-S. Yang (Ed.), *Nature-Inspired Optimization*

Algorithms (pp. 111-127). Oxford: Elsevier.

Yelghi, A., & Köse, C. (2018). A modified firefly algorithm for global minimum optimization. *Applied Soft Computing, 62*, 29-44.

Yu, S., Su, S., Lu, Q., & Huang, L. (2014). A novel wise step strategy for firefly algorithm. *International Journal of Computer Mathematics, 91*, 2507-2513.

Yu, S., Zhu, S., Ma, Y., & Mao, D. (2015a). Enhancing firefly algorithm using generalized opposition-based learning. *Computing, 97*, 741-754.

Yu, S., Zhu, S., Ma, Y., & Mao, D. (2015b). A variable step size firefly algorithm for numerical optimization. *Applied Mathematics and Computation, 263*, 214-220.

Zhou, L., Ding, L., Ma, M., & Tang, W. (2019). An accurate partially attracted firefly algorithm. *Computing, 101*, 477-493.

Table captions

Table 1. The list of main variants of FA between 2010 till 2019

Table 2. The list of functions and search results for YYFA behavior simulations

Table 3. CEC 2013 benchmark functions

Table 4. Parameter settings of each algorithm in comparison

Table 5. Results on the 10*D* benchmark functions

Table 4. Results on the 30*D* benchmark functions

Table 5. Results on the 50*D* benchmark functions

Table 8. Results of Wilcoxon signed rank test

Table 9. Effect of algorithm parameters

Table 10. Performance on constrained engineering optimization problems

Table 11. Rainfall intensity data of different recurrence periods and durations

Table 12. Comparison of fitting results between YYFA and FA

Figure captions

Figure 1. Pseudo code of FA

Figure 2. Probability distribution curves of different shapes and positions

Figure 3. Comparison of point distribution generated by GNS and random method in two-dimensional unit space

Figure 4. Algorithm of Firefly Moving in YYFA

Figure 5. Pseudo code of YYFA

Figure 6. Behavior of fireflies in YYFA for searching the global optimum

Figure 7. Convergence curves for the 10*D* case

Figure 8. Convergence curves for the 30*D* case

Figure 9. Convergence curves for the 50*D* case

Figure 10. Convergence curves for different parameter combinations