# Proof Central

Dear Author

Please use this PDF proof to check the layout of your article. If you would like any changes to be made to the layout, you can leave instructions in the online proofing interface. Making your changes directly in the online proofing interface is the quickest, easiest way to correct and submit your proof. Please note that changes made to the article in the online proofing interface will be added to the article before publication, but are not reflected in this PDF proof.

If you would prefer to submit your corrections by annotating the PDF proof, please download and submit an annotatable PDF proof by clicking here and you'll be redirected to our PDF Proofing system.
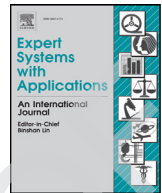
An improved firefly algorithm for global continuous optimization problems

Q1 Jinran Wu[a], You-Gan Wang[a,*], Kevin Burrage[a,b], Yu-Chu Tian[c], Brodie Lawson[a], Zhe Ding[c]

[a] *School of Mathematical Sciences, Queensland University of Technology, Australia*
[b] *Department of Computer Science, University of Oxford, UK*
[c] *School of Computer Science, Queensland University of Technology, Australia*

ARTICLE INFO

ABSTRACT

Global continuous optimization is populated by its implementation in many real-world applications. Such optimization problems are often solved by nature-inspired and meta-heuristic algorithms, including the firefly algorithm (FA), which offers fast exploration and exploitation. To further strengthen FA's search for global optimum, a Levy-flight FA (LF-FA) has been developed through sampling from a Levy distribution instead of the traditional uniform one. However, due to its poor exploitation in local areas, the LF-FA does not guarantee fast convergence. To address this problem, this paper provides an adaptive logarithmic spiral-Levy FA (AD-IFA) that strengthens the LF-FA's local exploitation and accelerates its convergence. Our AD-IFA is integrated with logarithmic-spiral guidance to its fireflies' paths, and adaptive switching between exploration and exploitation modes during the search process. Experimental results show that the AD-IFA presented in this paper consistently outperforms the standard FA and LF-FA for 29 test functions and 6 real cases of global optimization problems in terms of both computation speed and derived optimum.

© 2020 Published by Elsevier Ltd.

1. Introduction

Many engineering and business optimization problems are regarded as global optimization problems with many local optima. For example, the design of electronic systems, vehicle routine planning in large-scale traffic networks, the inverse problem of chemical kinetics and gene recognition in bioinformatics. In order to solve these optimization problems, two classes of methods have been developed: cooperative co-evolution (CC) methods (Potter, 1997) and non-decomposition methods (Hoang, 2008). The CC methods decompose the global continuous optimization to multiple low-dimensional sub-components. They can be further divided into two categories: static grouping based CC methods and dynamic grouping based CC methods (Chen, Johansson, Hsu, Liao, & Lin, 2016; Potter & De Jong, 1994). However, CC methods demand high computation costs, especially in non-linear continuous functions (Maeda, Fukano, Yamamichi, Nitta, & Kurata, 2011). Therefore, non-decompositional methods without a divide-and-conquer strategy is developed for improved computational efficiency. Among various non-decompositional methods, meta-heuristics algorithms become dominant with their strong exploration ability in global space. According to the nature-inspired origins, these meta-heuristic algorithms use two two main search methods: evolutionary computation and swarm intelligence. The evolutionary computation uses reputation, mutation, recombination and selection strategies to obtain the best offspring as the best optimum (Ashlock, 2006), as in genetic algorithm and differential evolution. The swarm intelligence imitates animals' behaviors for searching preys or other mates to update their positions (Banks, Vincent, & Anyakoha, 2007), as in particle swarm optimization (PSO) and ant colony optimization. Compared with evolutionary computation, swarm intelligence searches space faster (Nazir, Majid-Mirza, & Ali-Khan, 2014). To accelerate the speed for exploring and exploiting, inspired by the flashing patterns and behaviors of fireflies, Yang (2008) has firstly proposed the firefly algorithm (FA). According to Łukasik and Żak (2009), the FA performs better than PSO on continuous constrained optimization problems.

Due to its advantages, the FA has been widely used in many engineering applications. Senthilnath, Omkar, and Mani (2011) applied the FA for searching for the centers of the clusters by minimizing the distance sum of the patterns to their centers. Yang, Hosseini, and Gandomi (2012) proposed a FA method to address the economic dispatch problem for practical power systems

* Corresponding author.
E-mail addresses: jinran.wu@hdr.qut.edu.au (J. Wu), you-gan.wang@qut.edu.au (Y.-G. Wang), kevin.burrage@qut.edu.au (K. Burrage), y.tian@qut.edu.au (Y.-C. Tian), b.lawson@qut.edu.au (B. Lawson), zhe.ding@hdr.qut.edu.au (Z. Ding).

management. Kavousi-Fard, Samet, and Marzbani (2014) used the FA in searching the best parameter in support vector regression for accurate short load forecasting. Wang et al. (2012) introduced the FA to path planning for uninhabited combat air vehicles. In addition, according to Sayadi, Ramezanian, and Ghaffari-Nasab (2010), FA was also popularly used to solve NP-hard scheduling problems, traveling salesman problems (Jati et al., 2011) and digital image compression (Horng, 2012). For parameter estimation in expert systems, Sánchez, Melin, and Castillo (2017) introduced the FA to the modular granular neural networks for ear recognition and face recognition. Langari, Sardar, Mousavi, and Radfar (2020) incorporated FA to fuzzy clustering to protect the anonymized database and minimize the information loss. For online social networks, Jain and Katarya (2019) used the FA to discover the opinion lead in local communities. Moreover, Yang and He (2013) discussed two main reasons for the effectiveness of FA: the intelligent subdivision and the ability of dealing with multi-modality. From the inspiration of the dependence of FA on the variation of light intensity and their attractiveness, these two factors actually determine the searching performance (Fister, Fister Jr, Yang, & Brest, 2013).

Furthermore, to improve the performance of FA, some parameter adjustment mechanisms, e.g., fuzzy controllers, are incorporated, which can update the systemic parameters in FA based on the search process (Castillo, Soto, & Valdez, 2018; Lagunes, Castillo, Valdez, Soria, & Melin, 2018). The search path of the firefly is investigated for enhancing the exploration and exploitation of the fireflies. In 2010, Yang (2010a) introduced a Lévy flight move strategy to the FA, called the Lévy-flight FA (LF-FA). While LF-FA has a high probability of jumping out of a local minima, exploitation in local space is not promoted. Therefore, there are three primary **motivations** of our FA design in this paper:

a. A new position update path can be developed to enhance the exploitation in local space;
b. An intelligent controller based on the search process is required to maintain the balance between the exploration and the exploitation; and
c. For the global continuous optimization problems, a more efficient firefly algorithm is demanded with less computational cost and a better optimum.

In this paper, two novel modifications to FA are developed for continuous global optimization problems. The first one is that the logarithmic spiral path is proposed to improve the exploitation of fireflies in local space. The main idea is to combine this new path with light intensity and attractiveness to strengthen the exploitation based on the nature of the logarithmic spiral (Tamura & Yasuda, 2011). Then, an adaptive switch (ratio) is presented based on the searching dynamic process to determine the task mode: exploration in global space or exploitation in local space. As the switch is more sensitive to the local optimum, the ratio will dramatically auto-decrease to trigger the exploration mode during any potential trap. When the ratio is large, the exploitation mode will be tested. Combining these two designs gives an improved FA, which we referred to as AD-IFA.

This paper makes three main **contributions** as follows:

1) A logarithmic-spiral path is designed to improve the exploitation of search agents for local searching, leading to a speed-up of the FA convergence;
2) An adaptive switch (ratio) is proposed to balance the exploitation and the exploration during the search process. According to the variation of current fitness function value, the search agent can adapt its strategy to update the next position;
3) A novel adaptive logarithmic spiral-Lévy FA is developed for a more stable and high accurate optimal result;

In order to demonstrate our AD-IFA, 29 benchmark functions will be utilized (Surjanovic & Bingham, 2017; Yelghi & Köse, 2018). Furthermore, our AD-IFA is also tested in 6 real engineering applications to show its superiority to existing FA methods. The source code of our AD-IFA is available at: https://github.com/wujrtudou/AdaptiveFireflyAlgorithm.git

Tthis paper is organized as follows: Firstly, related work on FA and the Lévy-flight FA are reviewed in Section 2. Section 3 outlines the logarithmic spiral path, the adaptive switch (ratio) and our AD-IFA. Then, experimental settings, benchmark functions and simulation results are given in Section 4. Section 5 shows the efficiency of our AD-IFA through six engineering applications. Finally, Section 6 concludes the paper.

2. Related work

In this section, the FA and LF-FA will be outlined. In addition, the capacity of these two meta-heuristics optimization algorithms will also be explored.

2.1. The Firefly algorithm

As with other swarm intelligence optimization algorithms, the process of searching for a global optimum can be regarded as the consequence of the position change of search agents in a swarm (Yang, 2009). According to Yang (2010b), the FA stems from the idealized behavior of the flashing characteristics of fireflies when they fly to each other in the dark. For simplicity in describing FA, the following three rules have been proposed to idealize the behaviors of fireflies (Yang, 2010c):

a. All fireflies are unisex. So each firefly can be attracted by other fireflies regardless of sex;
b. The attractiveness of a firefly is decided by its brightness. So for any two flashing fireflies, the less brighter one tends to fly towards the brighter one. The attractiveness is proportional to the brightness. The brightness and the attractiveness both decrease with the increase of distance. The brightest firefly will fly randomly;
c. The brightness value for a firefly is determined by the landscape of the objective function.

In general, the brightness is assumed to be simply proportional to the objective function when dealing with the maximum problem. A minimum problem can be easily converted to a maximum problem.

In the FA framework, there are two issues: the variation of the light intensity and the formulation of the attractiveness. Generally speaking, the attractiveness of a firefly is determined by its brightness (or light intensity) associated with the objective function (Gazi & Passino, 2004). Meanwhile, the attractiveness is observed (judged) by eyes of fireflies. This means that the light intensity decreases with the distance from the source since light is absorbed in the media. So the attractiveness can be allowed to vary with the degree of absorption. Thus, the light intensity $I(r)$ is obtained as the inverse square law and absorption as follows:

$$I(r) = I_0 \cdot e^{-\gamma \cdot r^2}, \tag{1}$$

where I_0 and γ stand for the original light intensity and light absorption coefficient, respectively. Subsequently, the definition of attractiveness $\beta(r)$ is formulated by:

$$\beta(r) = \beta_0 \cdot e^{-\gamma \cdot r^2}, \tag{2}$$

where β_0 is the attractiveness when $\gamma = 0$. In Eq. 2, the distance between any two fireflies i and j at $\mathbf{x_i}$ and $\mathbf{x_j}$ is calculated by the
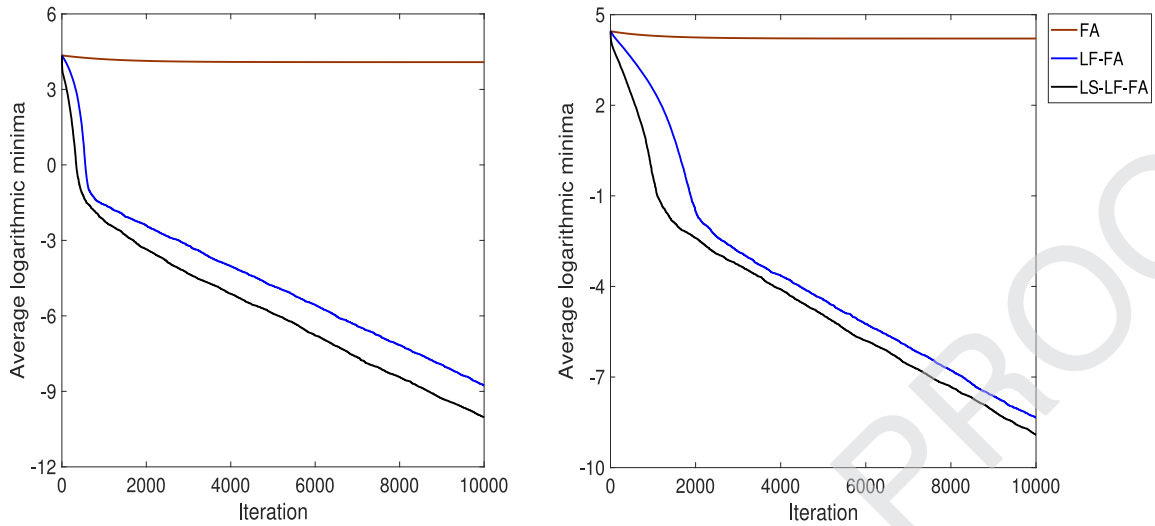
Fig. 1. The convergence curve of average $log_{10}(minima)$: Sphere function (left sub-figure) and Schwefel 1.2 function (right sub-figure)

Cartesian distance as follows:

$$r_{ij} = \sqrt{\sum_{p=1}^{d}(x_{i,p} - x_{j,p})^2}, \tag{3}$$

where $x_{i,p}$ is the p-th component of the spatial coordinate $\mathbf{x_i}$ of i-th firefly.

Let **rand** denote a d-dimensional uniform random vector in [0, 1]^d, and α be a parameter in [0,1], respectively. It follows that the new position $\mathbf{x_i}$ of a firefly, which is attracted by the brighter firefly j at time $t+1$, can be updated by:

$$\mathbf{x_{i,t+1}} = \mathbf{x_{i,t}} + \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \cdot (\mathbf{x_{j,t}} - \mathbf{x_{i,t}}) + \alpha \cdot (\mathbf{rand} - \mathbf{0.5}), \tag{4}$$

If the distance r is small, the random term avoids traps in the local minima. If r is large, the firefly behaves like a random walk.

FA has been shown to be a powerful method for non-linear design optimization problems (Koziel & Yang, 2011). Compared with existing PSO and evolutionary algorithms, FA has better performance in searching the global optimum (Yang, 2010b).

2.2. The Lévy-flight firefly algorithm

When addressing global continuous optimization problems, especially for high-dimensional optimization, the FA is often trapped in local optimum (Yang & He, 2013). Inspired by the research from Reynolds and Frye on the behavior of fruit flies, these insects can explore the landscape through a series of straight flight paths punctuated by a sudden 90° turn. This creates a Lévy-flight-style intermittent scale-free search pattern (Reynolds & Frye, 2007). Considering this inspiration, Yang (2010a) introduces the Lévy flight to FA in 2010, called the Lévy-flight firefly algorithm (LF-FA). LF-FA enhances the ability for exploration of the global space, while retaining some of FA's ability for exploitation in local space. It mainly applies the randomization from Lévy distribution instead of traditional uniform distribution. So the updated position Eq. 4 is modified to (Yang & He, 2013):

$$\mathbf{x_{i,t+1}} = \mathbf{x_{i,t}} + \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \cdot (\mathbf{x_{j,t}} - \mathbf{x_{i,t}}) + \alpha \cdot \text{sign}(\mathbf{rand} - \mathbf{0.5}) \otimes \mathbf{Levy}, \tag{5}$$

where α is the randomization parameter, and $\otimes$ is the Hadamard product. The term $\text{sign}(\mathbf{rand} - \mathbf{0.5})$ provides a random direction while the random step length is drawn from the Lévy flights. This modification significantly improves the probability of LF-FA to

jump out of the local optimum. Thus, it enhances the global search capability of the original FA (Yang & Deb, 2010).

The random number conforming the Lévy distribution is obtained as follows,

$$Levy(\eta) \sim \mu = t^{-1-\eta}, (0 \le \eta \le 2). \tag{6}$$

Here, the Levy random number is calculated by,

$$Levy(\eta) \sim \frac{\phi \times \mu}{|v|^{1/\eta}}, \tag{7}$$

where v and μ conform the standard normal distributions, and ϕ is calculated as follows,

$$\phi = \left[\frac{\tau(1+\eta) \times \sin(\pi \times \eta/2)}{\tau((1+\eta)/2) \times \eta \times 2^{(\eta-1)/2}}\right]^{1/\eta}, \tag{8}$$

where τ is standard Gamma function, and $\eta = 1.5$.

It is demonstrated that, compared with GA and PSO, LF-FA shows a higher success rate and less computational time in finding global optima (Yang & He, 2013). The main difference between FA and LF-FA is the position update equation, i.e., Eq. 4 for FA and Eq. 5 for LF-FA.

3. An improved firefly algorithm

This section starts with an exploration of the inspiration for logarithmic spiral paths. Then, an adaptive switch (ratio) is designed for the selection of searching methods to balance the exploration and exploitation based on the change in global fitness function values. After that, to combine with the new adaptive switch (ratio), an improved FA is presented, which we refer to as AD-IFA.

3.1. Design of the logarithmic spiral path

Although LF-FA strengthens the exploration of global space, it breaks the balance between exploration and exploitation. To clearly demonstrate this effect, the Sphere function $f(x) = \sum_{i=1}^{d} x_i^2$ and Schwefel 1.2 function $f(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j\right)^2$ (d is set as 10, and x_i ranges from -100 to 100) are selected as benchmark functions. The convergence curves of the average logarithmic minima are shown in Fig. 1. In our studies, experiments are executed 50 times to calculate the average performance. This can eliminates the impact of the randomness from the initialization of population position on the final results. In these 50 runs, all corresponding parameters are

set as the same value for each algorithm. The number of search agents is 15, $\alpha = 0.2$, $\beta_0 = 1$ and $\gamma = 1$.

It is seen from Fig. 1 that for both benchmark functions, the average logarithmic minimal values from LF-FA are smaller than those from FA. This shows that LF-FA strengthens the exploration in global space, and assists the search agents in jumping out of local traps. However, the ability of exploitation in local space has been ignored. How to maintain the exploration and exploitation for fireflies in the searching process is an interesting problem. Solving this problem will improve the efficiency of the optimizer and at the same time reduce the computational demand.

Inspired by the experimental biology, we address this dilemma by considering large birds of prey, such as peregrine falcons, which initially fly in a path similar to that of a logarithmic spiral in search of food (Tucker, 2000; Tucker, Tucker, Akers, & Enderson, 2000). We have also observed a similar path for fireflies during night. In addition, the whale optimization algorithm with a logarithmic spiral can actually strengthen the capability of exploitation in local space (Mirjalili & Lewis, 2016). The logarithmic spiral (LS) path is a potential tool to improve the exploitation for the FA. A new position update method by the logarithmic spiral is designed as follows:

$$\mathbf{x_{i,t+1}} = \mathbf{x_{i,t}} + \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \cdot (\mathbf{x_{j,t}} - \mathbf{x_{i,t}}) \otimes e^{b \cdot \mathbf{l}} \otimes \cos(2\pi \cdot \mathbf{l}). \tag{9}$$

In Eq. 9, $\mathbf{l}$ is a d-dimensional uniform random vector in $[-1, 1]^d$; b is a constant with a default value 1 for defining the shape of the logarithmic spiral. To explain the position update formulation in Eq. 9, this position variation is determined by two factors: brightness intensity and logarithmic spiral paths. The latter is mathematically expressed by the coefficients $e^{b \cdot \mathbf{l}} \otimes \cos(2\pi \cdot \mathbf{l})$.

3.2. Design of an adaptive switch

To balance the exploration mode (Levy Flight) and the exploitation mode (Logarithmic spiral), an adaptive switch (ratio) method is proposed in this paper to determine which method is applied in the current iteration,

$$\begin{cases} \text{Exploration Mode for Global Search,} & if & u > R_t, \\ \text{Exploitation Mode for Loal Search,} & if & u \le R_t, \end{cases} \tag{10}$$

where u is a uniform random number in [0,1] and the R_t is computed in the last iteration. To speed up the convergence of the optimizer, compared with exploration mode, the exploitation mode needs to be selected with a higher probability. So, we define the switch R_{t+1} ranging from [0.5, 1]. Setting the initial ratio as 0.5, we have the following adaptive switch value,

$$R_{t+1} = \begin{cases} \dfrac{1}{1 + \exp\left(-\frac{f_t^*}{f_{t-1}^*}\right)}, & \lfloor \lg|f_t^*| \rfloor \ne \lfloor \lg|f_{t-1}^*| \rfloor, \\ \dfrac{1}{1 + \exp\left(-\frac{f_t^* - \theta \cdot \lfloor \frac{f_t^*}{\theta} \rfloor}{f_{t-1}^* - \theta \cdot \lfloor \frac{f_{t-1}^*}{\theta} \rfloor}\right)}, & else. \end{cases} \tag{11}$$

In Eq. 11, f_t^* is the best fitness function value at the t-th iteration; $\lg(\cdot) = \log_{10}(\cdot)$; $\lfloor \cdot \rfloor$ is the floor function. The adaptive scale parameter threshold θ is calculated as follows,

$$\theta = 10^{\lfloor \lg|f_t^* - f_{t-1}^*| \rfloor + 1}. \tag{12}$$

To explore the role of this formula, we discuss three conditions as follows,

a. The first condition, $f_t^* \ggg f_{t-1}^*$. This means that there is a significant jump between two iterations. For the optimization process, search agents have found a new solution with distinct performance. So, the adaptive ratio R_{t+1} becomes 1, and the exploitation mode is chosen for next iteration;

b. If $f_t^* \lll f_{t-1}^*$, then in terms of performance degrading, switch to exploration. According to Eq. 11, the adaptive ratio R_{t+1} will be 0.5, and we will increase the probability of applying the exploration mode for the next search.

c. The last condition is $\lfloor \lg|f_t^*| \rfloor = \lfloor \lg|f_{t-1}^*| \rfloor$, implying that a local minima has been found. We modify the ratio by the item $\dfrac{f_t^* - \theta \cdot \lfloor \frac{f_t^*}{\theta} \rfloor}{f_{t-1}^* - \theta \cdot \lfloor \frac{f_{t-1}^*}{\theta} \rfloor}$ to make the adaptive switch more sensitive. In this case, search agents will have a high probability of jumping out of potential traps.

The scale coefficient $\lfloor \lg|f_t^* - f_{t-1}^*| \rfloor$ can auto-recognize the search state. Thus, the convergence is further improved by our adaptive switch. Next, to map the variation to a probability, the logistic function is introduced. Finally, the adaptive switch (ratio) is established. From the idea of adaptive switch design, we find that this switch has good capacity: during the searching process under a poor convergence, search agents are more active to select the exploration mode so that the optimizer is able to find a better optimum.

3.3. The improved firefly algorithm

Now the new update position formula is shown as follows,

$$\mathbf{x_{i,t+1}} = \begin{cases} \mathbf{x_{i,t}} + \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \cdot (\mathbf{x_{j,t}} - \mathbf{x_{i,t}}) \\ + \alpha \cdot sign[\mathbf{rand} - \mathbf{0.5}] \otimes \mathbf{Levy}, & u > R_t, \\ \mathbf{x_{i,t}} + \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \cdot (\mathbf{x_{j,t}} - \mathbf{x_{i,t}}) \\ \otimes e^{b \cdot \mathbf{l}} \otimes \cos(2\pi \cdot \mathbf{l}), & u \le R_t. \end{cases} \tag{13}$$

Our improved firefly algorithm, adaptive logarithmic spiral-Lévy flight firefly algorithm (AD-IFA), combines the merit of LF-FA with the new logarithmic spiral path controlled by an intelligent adaptive switch. The pseudo code and the flowchart are presented in Fig. 2.

It is seen from the pseudo code that some changes have been made to the standard FA framework in order to combine our adaptive switch. The current fitness function value f_t^* is recorded to establish the adaptive switch. To the performance of the logarithmic spiral (LS) path and the capacity of the adaptive switch, we will focus on a conventional method that the switch is fixed as 0.5 (Mirjalili & Lewis, 2016). In this special condition, the optimizer is named as the logarithmic spiral-Lévy flight firefly algorithm (LS-LF-FA), and the position update method is as follows,

$$\mathbf{x_{i,t+1}} = \begin{cases} \mathbf{x_{i,t}} + \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \cdot (\mathbf{x_{j,t}} - \mathbf{x_{i,t}}) \\ + \alpha \cdot sign[\mathbf{rand} - \mathbf{0.5}] \otimes \mathbf{Levy}, & u > 0.5, \\ \mathbf{x_{i,t}} + \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \cdot (\mathbf{x_{j,t}} - \mathbf{x_{i,t}}) \\ \otimes e^{b \cdot \mathbf{l}} \otimes \cos(2\pi \cdot \mathbf{l}), & u \le 0.5. \end{cases} \tag{14}$$

4. Numerical simulations

In this section, the improved firefly algorithm is tested on 29 test functions. More specifically, the logarithmic spiral and the adaptive switch will be tested, respectively. Nine benchmark functions are selected for analysis of our new optimizer. Twenty additional functions and their simulation results are recorded in Appendix A.

4.1. Experimental settings

The experimental settings for the parameters in all firefly algorithms (FA, LF-FA, LS-LF-FA, and our AD-IFA) are as follows: the randomization parameter $\alpha = 0.2$, the fixed light absorption coefficient $\gamma = 1$ and the attractiveness at $r = 0$ is $\beta_0 = 1$. Moreover,

Algorithm 2 The Improved firefly algorithm AD-IFA

The fitness function $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \cdots, x_d)^T$

Initialize a population of fireflies $\mathbf{x_i}(i = 1, 2, \cdots, n)$

Set the initial ratio as 0.5

Definite light absorption coefficient γ

while $t <$ MaxGeneration **do**

  for $i = 1 : n$ all n fireflies **do**

    for $j = 1 : n$ all n fireflies (inner loop) **do**

      Light intensity ℓ_i at $\mathbf{x_i}$ is determined by $f(\mathbf{x_i})$

      if $\ell_i < \ell_j$ **then**

        if $u > R_t$ **then**

          Update the i-th firefly's position using Eq. 5

        else

          Update the i-th firefly's position using Eq. 9

        end if

      end if

      Attractiveness varies with distance r via $\exp(-\gamma r^2)$

      Evaluate new solutions and update light intensity

    end for

  end for

  Rank the fireflies and find the current best fitness function value f_l^*

  Update the current ratio R_{t+1} using Eq. 11

end while

Output results and visualisation.
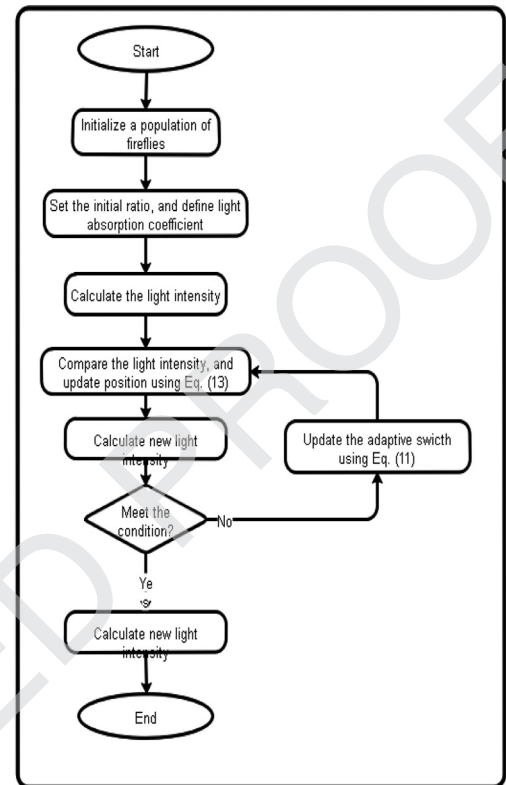
Fig. 2. The pseudo code and flowchart of our AD-IFA.

Table 1
Benchmark functions I.

Function	Equation description
Cross-in-tray	$f_1(x) = -0.0001\left(\left\|\sin(x_1)\sin(x_2)\exp\left(\left\|100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right\|\right)\right\| + 1\right)^{0.1}$, $x_i \in [-10, 10]$
Schaffer N.2	$f_2(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$, $x_i \in [-100, 100]$
Bohachevsky	$f_3(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$, $x_i \in [-100, 100]$
Six-hump camel	$f_4(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2$, $x_1 \in [-3, 3], x_2 \in [-2, 2]$
Ackley	$f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + \exp(1)$, $x_i \in [-32, 32]$
Rotated hyper-ellipsoid	$f_6(x) = \sum_{i=1}^{d}\sum_{j=1}^{i}x_j^2$, $x_i \in [-65.536, 65.536]$
Sum of different powers	$f_7(x) = \sum_{i=1}^{d}\|x_i\|^{i+1}$, $x_i \in [-1, 1]$
Zakharov	$f_8(x) = \sum_{i=1}^{d}x_i^2 + \left(\sum_{i=1}^{d}0.5ix_i\right)^2 + \left(\sum_{i=1}^{d}0.5ix_i\right)^4$, $x_i \in [-5, 10]$
Tablet	$f_9(x) = 10^6 \cdot x_1^2 + \sum_{i=2}^{d}x_i^2$, $x_i \in [-1, 1]$

considering the test function with different dimensions in the experiments, we set two parameters to halt experiments. The first one is the tolerance $\epsilon = 10^{-4}$ for all experiments regardless of the dimensions of the test functions. The other one is the number of maximum iterations *Max_iteration* decided by their dimension as follows. It is set as 300,000 for 2-dimensional functions, 400,000 for 8, 16 and 64-dimensional functions, and 450,000 for 100-dimensional functions, respectively. Once either condition is satisfied, the experiments are immediately terminated. Furthermore, the numbers of search agents *num_search* for 2, 8, 16, 64 and 100-dimensional functions are set as 15, 25, 30, 35 and 40, respectively. We run each experiment 50 times to eliminate the

impact of the initial parameters on the calculation results. For fair comparisons among all tested approaches, we use the same initial agent positions for each sub-experiment.

4.2. Benchmark functions

Benchmark functions are taken from the Virtual Library of Simulation Experiments: Test Functions and Datasets (Surjanovic & Bingham, 2017) and the research by Yelghi and Köse (2018) in Table 1 and 4. These benchmark functions have different shapes and characteristics, including many local minima, bowl-shaped, plate-shaped, valley-shaped, and steep ridges/drops.

Table 2
Results and comparison I.

function	FA		LF-FA		LS-LF-FA		AD-IFA	
	success rate	iteration	success rate	iteration	success rate	iteration	success rate	iteration
d=2								
f_1	48%	27(1.00)	100%	54(2.00)	100%	39(1.44)	100%	34(1.26)
f_2	8%	2741(1.00)	100%	395(0.14)	100%	186(0.07)	100%	203(0.07)
f_3	100%	1311(1.00)	100%	4658(3.55)	100%	333(0.25)	100%	177(0.14)
f_4	88%	118(1.00)	100%	1389(11.77)	100%	160(1.36)	100%	59(0.50)
d=8								
f_5	0%	400000(1.00)	100%	326837(0.82)	100%	38977(0.10)	100%	13025(0.03)
f_6	100%	100421(1.00)	100%	141338(1.41)	100%	5497(0.05)	100%	2524(0.03)
f_7	100%	125(1.00)	100%	4460(35.68)	100%	413(3.30)	100%	223(1.78)
f_8	100%	77130(1.00)	100%	129067(1.67)	100%	5143(0.07)	100%	2478(0.03)
f_9	100%	116700(1.00)	100%	176046(1.51)	100%	28336(0.24)	100%	8714(0.07)
d=16								
f_5	0%	400000(1.00)	100%	365389(0.91)	100%	84987(0.21)	100%	35329(0.09)
f_6	100%	157818(1.00)	100%	209503(1.33)	100%	24915(0.16)	100%	10763(0.07)
f_7	100%	383(1.00)	100%	15267(39.86)	100%	865(2.26)	100%	374(0.98)
f_8	100%	124276(1.00)	100%	178011(1.43)	100%	21255(0.17)	100%	9609(0.77)
f_9	100%	154758(1.00)	100%	220863(1.43)	100%	54748(0.35)	100%	19348(0.13)
d=64								
f_5	0%	400000(1.00)	0%	400000(1.00)	100%	225671(0.56)	100%	161177(0.40)
f_6	100%	283206(1.00)	100%	328261(1.16)	100%	211360(0.75)	100%	156321(0.55)
f_7	100%	3508(1.00)	100%	52909(15.08)	100%	1023(0.29)	100%	468(0.13)
f_8	100%	203399(1.00)	100%	284709(1.40)	100%	160412(0.79)	100%	98452(0.05)
f_9	100%	227089(1.00)	100%	303246(1.34)	100%	177544(0.78)	100%	80387(0.35)
d=100								
f_5	0%	450000(1.00)	0%	450000(1.00)	100%	294437(0.65)	100%	223782(0.50)
f_6	100%	380307(1.00)	100%	411923(1.08)	100%	343818(0.90)	100%	281291(0.74)
f_7	100%	8697(1.00)	100%	74215(8.53)	100%	893(0.10)	100%	454(0.05)
f_8	100%	259029(1.00)	100%	359475(1.39)	100%	261647(1.01)	100%	184103(0.71)
f_9	100%	279258(1.00)	100%	368817(1.32)	100%	245720(0.88)	100%	129625(0.46)

Number format: the average number of iterations (the ratio of the average number of iterations to the average number of iterations by FA).

4.3. Analysis of simulation results

Experimental results are recorded in Tables 2 and 5. In evaluating simulation results, we define "success" when the optimizer meets the following two requirements: a). the value of the best position is smaller than 10^{-4}; and b). the number of iterations is less than the set maximum number of iterations. We will count all "success" and calculate the success rate. If the optimizer successfully finds the global optimum, the corresponding number of iterations is recorded to compute the average number of iterations ("*iteration*" in all tables).

4.3.1. Analysis of the Logarithmic spiral path: LS-LF-FA

As illustrated in Table 2, when searching the optimum for low-dimensional benchmark functions, we have the following observations:

a. When comparing FA with LF-FA, the LF-FA optimizer achieves higher success rates as it strengthens exploration in the global space. One of the obvious examples is for f_1, the success rate for increases from 48% to 100%. However, when we analyse f_3, although both optimizer show a success rate of 100%, the computational demand of LF-FA is more than that of FA. For example, for f_3, the average number of iterations for LF-FA is 4658 while it is only 1311 for FA;

b. When focusing on the results from LF-FA and LS-LF-FA, we find LS-LF-FA has the potential to reduce the number of iterations with a high success rate. For instance, for the 100-dimensional f_7, although both optimization algorithms have the same success rate (100%), LS-LF-FA saves around 98.80% computational cost. This means the logarithmic spiral path significantly addresses the poor convergence resulting from the Lévy-flight path;

c. The last point is to compare the original FA with LS-LF-FA. It is observed from the test results that LS-LF-FA has achieved satis-

factory performance with high success rates and less computation cost. This is because the Lévy flight improves the exploration in the global space and the success rate raises; the logarithmic spiral path strengthens the exploitation in the local space, and the number of iterations is reduced. As a result, LS-LF-FA can re-balance the exploration and exploitation. It also intensifies the capacity of optimizing by the search agent (firefly).

Furthermore, as shown in Table 2, the benchmark functions with 8, 16, 64 and 100 dimensions are recorded, respectively. With the increase in the dimension, the problems become more difficult. We find that the LS-LF-FA optimizer outperforms both FA family algorithms. Considering the shapes and characteristics of benchmark functions, we find that: a). for the uni-modal functions, LS-LF-FA performs the best, showing its good exploitation ability; b). for the multi-modal functions, LS-LF-FA provides sufficient global search to explore the large search space.

4.3.2. Analysis of the adaptive switch: our AD-IFA

From Table 2, compared with LS-LF-FA, our improved firefly algorithm AD-IFA saves the computational cost for both low- and high-dimensional optimization. For example, for the 2-dimensional optimization, searching for the optimum of the Bohachevsky function f_3 costs 333 iterations when using LS-LF-FA while our AD-IFA needs only 177 iterations. For another test function, Ackley function f_5 with 64 dimensions, our AD-IFA spends 161,177 iterations while 225,671 iterations are needed in LS-LF-FA.

To visibly show the searching process, two videos of the update positions of the search agent (firefly) are displayed in in Figs. 3 and 4, respectively, for two different 2-dimensional test functions (an extreme complex function f_2 and a simple function f_3). When glancing at Fig. 4, our AD-IFA (black dots), we observe there is still global search going on while some agents cluster in the middle
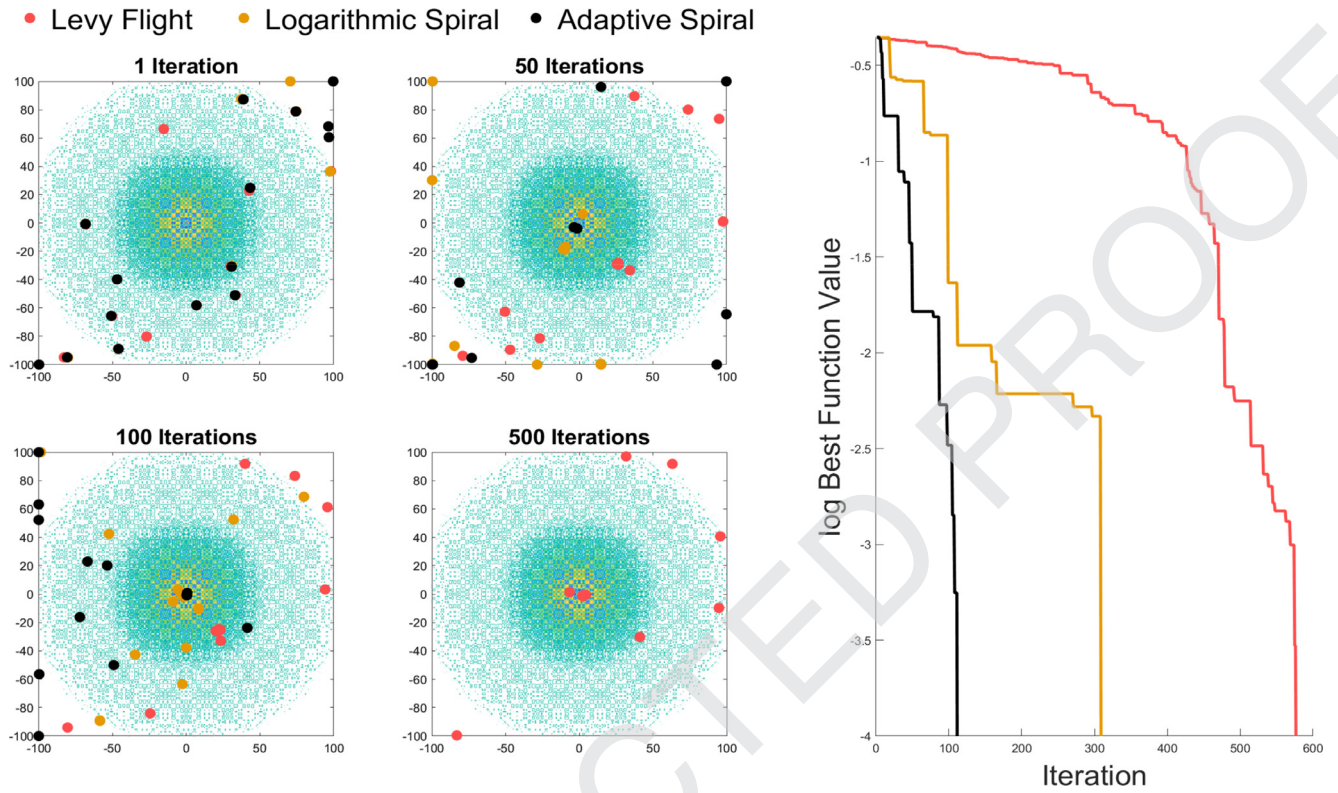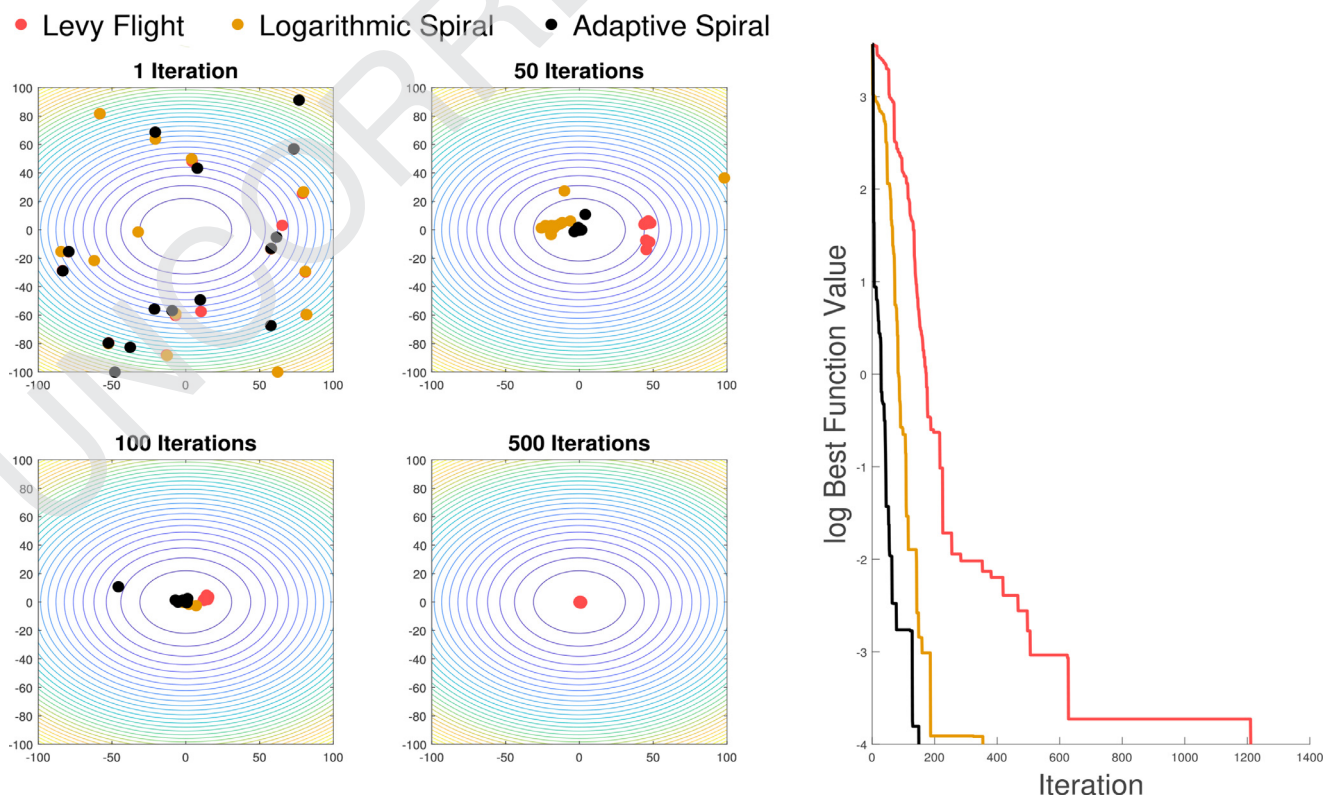
Fig. 3. The position update process and fitness function for the complex function f_2

Fig. 4. The position update process and fitness function for the simple function f_3

Table 3
Computational results for six cases.

| case	optimizer	Mean	Best	Std
cantilever beam				
	FA	8.72	3.59	1.93
	LF-FA	4.91	1.45	1.63
	LS-LF-FA	1.95	1.34	0.91
	AD-IFA	1.54	1.34	0.44
corrugated bulkhead design				
	FA	10.23	7.21	1.95
	LF-FA	8.83	6.95	1.26
	LS-LF-FA	7.44	6.86	0.67
	AD-IFA	7.21	6.84	0.58
pressure vessel design				
	FA	1031286.45	13808.08	956749.15
	LF-FA	174749.48	6299.77	188123.30
	LS-LF-FA	25954.57	5923.35	49408.94
	AD-IFA	10050.85	5886.65	8629.44
a three-bar truss design				
	FA	287.84	282.84	4.94
	LF-FA	283.20	282.84	1.10
	LS-LF-FA	282.84	282.84	0.00
	AD-IFA	282.84	282.84	0.00
tubular column design				
	FA	28.74	26.52	2.08
	LF-FA	27.46	26.50	1.36
	LS-LF-FA	26.58	26.50	0.19
	AD-IFA	26.54	26.50	0.07
welded beam design				
	FA	3.41	1.88	0.92
	LF-FA	3.13	1.93	0.80
	LS-LF-FA	2.81	1.85	0.63
	AD-IFA	2.40	1.81	0.50
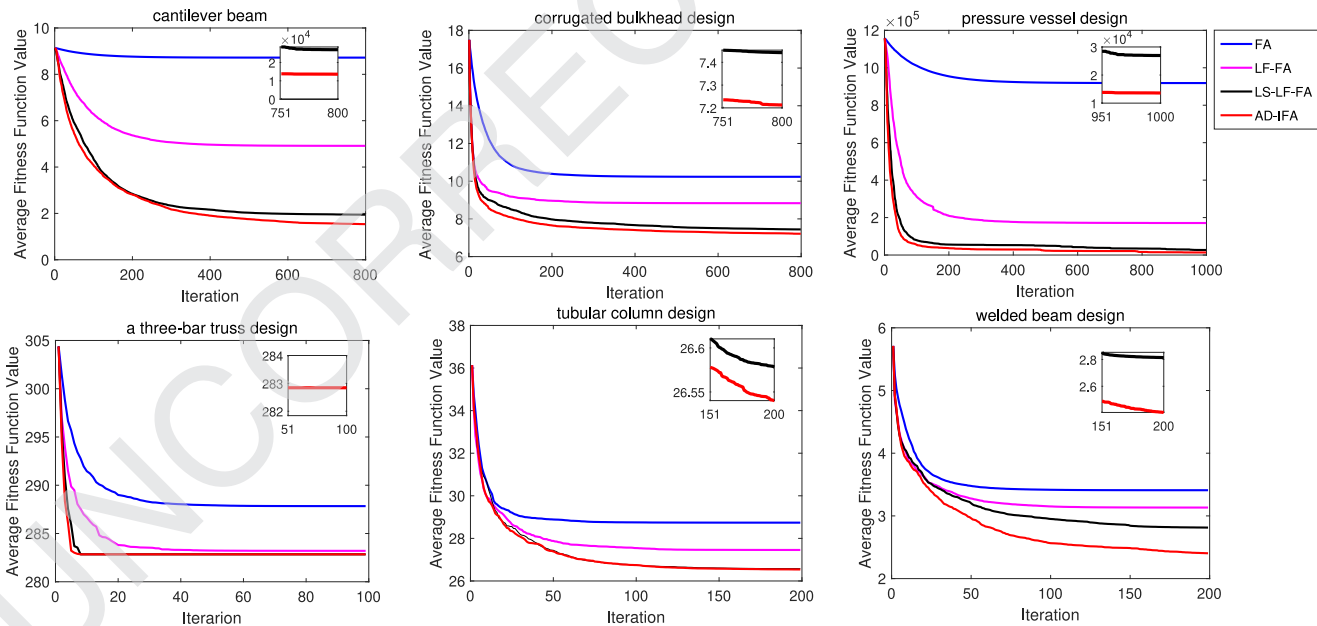
Fig. 5. The position update process and fitness function for all 6 cases

and search out the optimum. The whole position update process is recorded in Supplementary.

To summarize the above analysis, our improved firefly algorithm AD-IFA has a powerful exploitation ability to search local space and an efficient exploration capability to search the global space. The logarithmic spiral path enhances the exploitation in local space. It also improves the success rate and reduces the computational cost measured by the number of iterations. The second finding is that the adaptive switch (ratio) controls searching methods (exploitation mode or exploration mode) based on historical records from the searching process. As a result, the number of iterations is further reduced. Thus, our modification is a promising tool for global continuous optimization problems.

5. Case studies of real-world applications

In this section, the performance of the four mentioned firefly algorithms, FA, LF-FA, LS-LF-FA, and our AD-IFA, is evaluated in six engineering real applications (Gandomi, Yang, & Alavi, 2013): cantilever beam (Fleury & Braibant, 1986), corrugated bulkhead design (Moses & Onoda, 1969), pressure vessel design (Gandomi et al., 2013), a three-bar truss design (Nowacki, 1973), tubular column design (Rao, 2019), and welded beam design (Coello, 2000). Their

mathematical formulations are presented in Appendix B. The values of the parameters setting α, γ, β_0 for our four firefly algorithms are set at 0.2, 1, and 1, respectively. Respected to each scenario, the *Max_iteration* is set as 800, 800, 1000, 100, 200, and 200, for cantilever beam ($d = 5$), corrugated bulkhead design ($d = 4$), pressure vessel design ($d = 4$), a three-bar truss design ($d = 2$), tubular column design ($d = 2$), and welded beam design ($d = 4$) with 15 search agents, respectively. In addition, all experiments are repeated 50 times for performance evaluation. The Mean, the Best, and the standard deviation (Std) of the optimal results for each scenario at the maximum iteration are presented to show the efficiency of the optimizer (accuracy and robustness). The computational results are recorded in Table 3, and mean convergence curves for each scenario are displayed in Fig. 5.

As illustrated in Table 3, all optima found using our AD-IFA are superior with better stability to those from other algorithms. The mean, the best, and the standard deviation for 50 experiments using our AD-IFA are less than those using FA, LF-FA and LS-LF-FA in all scenarios. Compared the LF-FA and LS-LF-FA, our proposed logarithmic spiral path efficiently speeds up the convergence and a better optimum. For instance, in the corrugated bulkhead design, the Mean and the Best obtain 1.39 and 0.09 promotion, respectively, while the Std decreases from 1.26 to 0.67.

Meanwhile, the adaptive switch can significantly balance the exploration and the exploitation. In the pressure vessel design optimization, the Mean and the Std of results from our AD-IFA (Mean: 10050.85, and Std: 8629.44) are remarkably less than those from LS-LF-FA (Mean: 25954.57, and Std: 49408.94). Furthermore, the best result from our AD-IFA is better than that from LS-LF-FA. In addition, Fig. 5 displays the convergence curves corresponding to the results in Table 3. It is seen that he red curve from our AD-IFA is below the blue one from FA, pink from LF-FA and black one from LS-LF-FA in all the 6 cases during the search process, showing our AD-IFA is significantly more efficient with fast convergence and lower minimal values than other algorithms.

6. Conclusions

The original FA suffers from low exploration in global space and low exploitation in local space. The modified version, LF-FA, focuses on the exploration but results in slowed convergence because of poor exploitation in local space. In this research, the logarithmic-spiral path has been applied in the FA optimizer to strengthen the exploitation in local space and accelerate the convergence of the search process. Then, an adaptive switch has been created based on the current fitness function value variation to determine the search mode (exploration or exploitation) at next iteration. Based on comparative studies, we have found that our improved firefly algorithm has a higher success rate of finding the global optimum for different benchmark functions. It also reduces the computational cost. Our case studies have shown that the AD-IFA presented in this paper provides more stable, more reliable, and more accurate optimal results for global continuous optimization problems. Therefore, the new optimizer is promising to address NP-hard problems with less computational cost and improved solutions.

Although our AD-IFA can achieve significant improvements, there are still some **limitations** for our optimizer. The first one is that the computational cost is still relatively high using interactive position information of each agent to update the next position for each agent. The choice of parameters (the randomization parameter α, and the light absorption coefficient γ) will impact on the performance of our proposed optimization. Furthermore, from the results for f_{29} at $d = 100$, the success rate is quite low, indicating that the family of firefly optimization algorithms is weak in some problems.

In the future, following our research, a number of potential directions can be explored, which mainly are divided into two types: the mechanism of the swarm intelligence, and the specific real applications. For the mechanism exploration in swarm intelligence, our proposed logarithmic-spiral path can be incorporated to any optimization, e.g., particle swarm algorithm, and grey wolf optimizer. In addition, our adaptive switch can be developed in the swarm intelligence. For example, it can speed up the convergence by balancing exploration and exploitation in the whale optimization algorithm. To address the discussed limitation of parameter sensitivity, an adaptive mechanism can be investigated for parameter setting in firefly algorithms. Finally, due to its advantages, our AD-IFA can be further tailored for applications in practical systems, e.g., the privacy preservation in social networks, the multicast routing in vehicular communication, and parameter selection in machine learning model training.

Declaration of Competing Interest

All authors declare no conflict interest.

Credit authorship contribution statement

Jinran Wu: Conceptualization, Investigation, Methodology, Validation, Software, Writing - original draft. **You-Gan Wang:** Supervision, Funding acquisition, Project administration, Writing - original draft, Writing - review & editing. **Kevin Burrage:** Supervision, Methodology, Writing - review & editing. **Yu-Chu Tian:** Supervision, Writing - review & editing. **Brodie Lawson:** Formal analysis, Visualization, Writing - original draft, Writing - review & editing. **Zhe Ding:** Conceptualization, Writing - original draft, Writing - review & editing.

Acknowledgements

Appendix A.

To show the capacity of LS-LF-FA, 20 additional benchmark functions are listed in Table 3. The simulation results are recorded in Table 5.

Appendix B.

The detailed mathematical formulations in Section 5 are given as:

Case 1: Cantilever beam (Fleury & Braibant, 1986),

$$\begin{aligned} &\min \ f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5) \\ &\text{S.T. } g = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leqslant 0. \end{aligned} \quad (15)$$

where $0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$.

Table 4
Benchmark functions II.

Function	Equation description
Drop-wave	$f_{10}(x) = -\frac{1+\cos\left(12\sqrt{x_1^2+x_2^2}\right)}{0.5(x_1^2+x_2^2)+2}$, $x_i \in [-5.12, 5.12]$
Eggholder	$f_{11}(x) = -(x_2 + 47)\sin\left(\sqrt{\left|x_2 + \frac{x_1}{2} + 47\right|}\right) - x_1\sin\left(\sqrt{\left|x_1 - (x_2 + 47)\right|}\right)$, $x_i \in [512, 512]$
Matyas	$f_{12}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$, $x_i \in [-10, 10]$
Holder table	$f_{13}(x) = -\left|\sin(x_1)\cos(x_2)\exp\left(\left|1 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right|\right)\right|$, $x_i \in [-10, 10]$
Rastrgin	$f_{14}(x) = 20 + \sum_{i=1}^{2}[x_i^2 - 10\cos(2\pi x_i)]$, $x_i \in [-5.12, 5.12]$
Shubert	$f_{15}(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$, $x_i \in [-5.12, 5.12]$
Bohachevsky	$f_{16}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$, $x_i \in [-100, 100]$
	$f_{17}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$, $x_i \in [-100, 100]$
Booth	$f_{18}(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$, $x_i \in [-10, 10]$
Mccormick	$f_{19}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$, $x_1 \in [-1.5, 4], x_2 \in [-3, 4]$
Three-hump camel	$f_{20}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$, $x_i \in [-5, 5]$
Michalewicz	$f_{21}(x) = -\sum_{i=1}^{2}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$, $x_i \in [0, \pi]$
Goldstein-price	$f_{22}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]\dots$
	$\dots \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$, $x_i \in [-2, 2]$
Levy	$f_{23}(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{d-1}(\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_i + 1)] + (\omega_d - 1)^2[1 + \sin^2(2\pi\omega_d)]$,
	$\omega_i = 1 + \frac{x_i - 1}{4}$, $x_i \in [-10, 10]$
Sphere	$f_{24}(x) = \sum_{i=1}^{d} x_i^2$, $x_i \in [-5.12, 5.12]$
Sum squares	$f_{25}(x) = \sum_{i=1}^{d} ix_i^2$, $x_i \in [-10, 10]$
Styblinski-Tang	$f_{26}(x) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)$, $x_i \in [-5, 5]$
Step	$f_{27}(x) = \sum_{i=1}^{d}\lfloor x_i + 0.5\rfloor^2$, $x_i \in [-100, 100]$
Griewank	$f_{28}(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}}) + 1$, $x_i \in [-60, 60]$
Generalized schaffer	$f_{29}(x) = 0.5 + ((\sin(\sum_{i=1}^{d} x_i^2))^2 - 0.5)\cdot(1 + 0.001((\sum_{i=1}^{d} x_i^2))^{-2}$, $x_i \in [-100, 100]$

Case 2: Corrugated bulkhead design (Moses & Onoda, 1969),

$$\min\ f(b, h, l, t) = \frac{5.885t(b+l)}{b + \sqrt{l^2 - h^2}},$$

$$\text{s.t.}\ \begin{cases} g_1 = th\left(0.4b + \frac{l}{6}\right) - 8.94\left(b + \sqrt{l^2 - h^2}\right) \geqslant 0, \\ g_2 = th^2\left(0.2b + \frac{l}{12}\right) \\ \quad -2.2\left(8.94\left(b + \sqrt{l^2 - h^2}\right)\right)^{\frac{4}{3}} \geqslant 0, \\ g_3 = t - 0.0156b - 0.15 \geqslant 0, \\ g_4 = t - 0.0156l - 0.15 \geqslant 0, \\ g_5 = t - 1.05 \geqslant 0, \\ g_6 = t - h \geqslant 0. \end{cases}$$

(16)

where $0 \leq b, h, l \leq 100$, and $0 \leq t \leq 5$.

Case 3: Pressure vessel design (Gandomi et al., 2013),

$$\min\ f(T_s, T_h, R, L) = 0.6224T_sRL + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_h^2L,$$

$$\text{s.t.}\ \begin{cases} g_1 = -T_s + 0.0193R \leqslant 0, \\ g_2 = -T_h + 0.0095R \leqslant 0, \\ g_3 = -\pi R^2 L - \frac{4}{3}\pi R^3 + 1296000 \leqslant 0, \\ g_4 = L - 240 \leqslant 0. \end{cases}$$

(17)

where $1 \times 0.0625 \leq T_s, T_h \leq 99 \times 0.0625$, and $10 \leq R, L \leq 200$.

Case 4: A three-bar truss design (Nowacki, 1973),

$$\min\ f(A_1, A_2) = \left(2\sqrt{2A_1} + A_2\right)l,$$

$$\text{s.t.}\ \begin{cases} g_1 = \frac{\sqrt{2}A_1 + A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P - \sigma \leqslant 0, \\ g_2 = \frac{A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P - \sigma \leqslant 0, \\ g_3 = \frac{1}{A_1 + \sqrt{2}A_2}P - \sigma \leqslant 0. \end{cases}$$

(18)

with $l = 100cm$, $P = 2KN/CM^2$, and $\sigma = 2KN/CM^2$ $(0 \leq A_1 \leq 1$, and $0 \leq A_2 \leq 1)$.

Case 5: Tubular column design (Rao, 2019),

$$\min\ f(d, t) = 9.8dt + 2d,$$

$$\text{s.t.}\ \begin{cases} g_1 = \frac{P}{\pi dt\sigma_y} - 1 \leqslant 0, \\ g_2 = \frac{8PL^2}{\pi^3Edt(d^2 + t^2)} - 1 \leqslant 0, \\ g_3 = \frac{2.0}{d} - 1 \leqslant 0, \\ g_4 = \frac{d}{14} - 1 \leqslant 0, \\ g_5 = \frac{0.2}{t} - 1 \leqslant 0, \\ g_6 = \frac{t}{0.8} - 1 \leqslant 0. \end{cases}$$

(19)

with $P = 2500kgf$, $\sigma_y = 500kgf/cm^2$, $E = 0.85 \times 10^6kgf/cm^2$, and $L = 250cm$ $(2 \leq d \leq 14$ and $0.2 \leq t \leq 0.8)$.

Table 5
Results and comparison II.

		FA		LF-FA		LS-LF-FA	
dimension	function	success rate	iteration	success rate	iteration	success rate	iteration
d=2							
	f_{10}	68%	2324(1.00)	100%	6064(2.61)	100%	744(0.32)
	f_{11}	4%	1070(1.00)	4%	208(0.19)	16%	45(0.04)
	f_{12}	100%	106(1.00)	100%	63(0.59)	100%	46(0.43)
	f_{13}	44%	289(1.00)	100%	2351(8.13)	100%	212(0.73)
	f_{14}	0%	300000(1.00)	100%	20458(0.07)	100%	754(0.00)
	f_{15}	22%	20705(1.00)	100%	46216(2.23)	100%	1261(0.06)
	f_{16}	100%	1150(1.00)	100%	4792(4.17)	100%	331(0.29)
	f_{17}	100%	867(1.00)	100%	1676(1.93)	100%	267(0.31)
	f_{18}	100%	160(1.00)	100%	1146(7.16)	100%	134(0.84)
	f_{19}	76%	197(1.00)	100%	1799(9.13)	100%	233(1.18)
	f_{20}	62%	57(1.00)	100%	371(6.51)	100%	101(1.77)
	f_{21}	92%	651(1.00)	100%	5407(8.31)	100%	347(0.53)
	f_{22}	92%	5329(1.00)	100%	23130(4.34)	100%	637(0.12)
d=8							
	f_{23}	0%	400000(1.00)	100%	94987(0.24)	100%	3627(0.01)
	f_{24}	100%	71894(1.00)	100%	111418(1.55)	100%	3197(0.04)
	f_{25}	100%	636(1.00)	100%	448(0.70)	100%	135(0.21)
	f_{26}	2%	63762(1.00)	100%	116763 (1.83)	100%	5969(0.09)
	f_{27}	100%	10834(1.00)	100%	240(0.02)	100%	170(0.02)
	f_{28}	0%	400000(1.00)	0%	400000(1.00)	0%	400000(1.00)
	f_{29}	0%	400000(1.00))	0%	400000(1.00)	84%	106312(0.27)
d=16							
	f_{23}	0%	400000(1.00)	100%	148386(0.37)	100%	12846(0.03)
	f_{24}	100%	113313 (1.00)	100%	163247(1.44)	100%	9942(0.09)
	f_{25}	100%	30520(1.00)	100%	46764(1.53)	100%	428(0.01)
	f_{26}	0%	400000(1.00)	100%	151471(0.38)	100%	24251(0.06)
	f_{27}	100%	26987(1.00)	100%	767(0.03)	100%	380(0.01)
	f_{28}	0%	400000(1.00)	18%	113138 (0.28)	10%	6690(0.02)
	f_{29}	0%	400000(1.00)	0%	400000(1.00)	64%	156692(0.39)
d=64							
	f_{23}	0%	400000(1.00)	100%	236048(0.59)	100%	105978(0.26)
	f_{24}	100%	179317(1.00)	100%	251272(1.40)	100%	72983(0.41)
	f_{25}	100%	259972(1.00)	100%	293018(1.13)	100%	2961(0.01)
	f_{26}	0%	400000(1.00)	100%	210556(0.53)	100%	137016(0.34)
	f_{27}	100%	18556(1.00)	100%	23432(1.26)	100%	28(0.00)
	f_{28}	74%	105743(1.00)	72%	175488(1.66)	72%	46736(0.44)
	f_{29}	0%	400000(1.00)	0%	400000(1.00)	44%	248051(0.62)
d=100							
	f_{23}	0%	450000(1.00)	100%	296931(0.66)	100%	157685(0.35)
	f_{24}	100%	225889(1.00)	100%	312314(1.38)	100%	121198(0.54)
	f_{25}	100%	361676(1.00)	100%	378297(1.05)	100%	4660(0.01)
	f_{26}	0%	450000(1.00)	96%	256583(0.57)	82%	220995(0.49)
	f_{27}	100%	42685(1.00)	100%	14508(0.34)	100%	81(0.00)
	f_{28}	62%	134860(1.00)	66%	215805(1.60)	58%	89644(0.66)
	f_{29}	0%	450000(1.00)	0%	450000(1.00)	14%	340897(0.76)

Case 6: Welded beam design (Coello, 2000),

$$\min \ f(x) = 1.10471x_1^2 x_2 + 0.04811x_3 x_4 (14.0 + x_2),$$

$$\text{S.T.} \begin{cases} g_1 = \tau(x) - \tau_{\max} \leqslant 0, \\ g_2 = \sigma(x) - \sigma_{\max} \leqslant 0, \\ g_3 = x_1 - x_4 \leqslant 0, \\ g_4 = 0.10471x_1^2 + 0.04811x_3 x_4(14 + x_2) - 5 \leqslant 0, \\ g_5 = 0.125 - x_1 \leqslant 0, \\ g_6 = \delta(x) - \delta_{\max} \leqslant 0, \\ g_7 = P - P_c(x) \leqslant 0. \end{cases}$$

$$(20)$$

with $\quad \tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau'' \dfrac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \dfrac{P}{\sqrt{2}x_1 x_2},$

$\tau'' = \dfrac{MR}{J}, \quad M = P\left(L + \dfrac{x_2}{2}\right), \quad R = \sqrt{\left(\dfrac{x_2}{2}\right)^2 + \left(\dfrac{x_1 + x_3}{2}\right)^2},$

$J = 2\left(\sqrt{2}x_1 x_2 \left(\dfrac{x_2^2}{12} + \left(\dfrac{x_1 + x_3}{2}\right)^2\right)\right), \quad \sigma(x) = \dfrac{6PL}{x_3^2 x_4},$

$\delta(x) = \dfrac{4PL^3}{Ex_3^3 x_4}, \quad P_c(x) = \dfrac{4.013E}{L^2} \sqrt{\dfrac{x_3^2 x_4^6}{36}} \left(1 - \dfrac{x_3}{2L} \sqrt{\dfrac{E}{4G}}\right),$

$P = 6000$lb, $L = 14$in, $E = 30 \times 10^6$psi, $G = 12 \times 10^6$psi, $\tau_{\max} = 13600$psi, $\sigma_{\max} = 30000$psi, and $\delta_{\max} = 0.25$in ($0.1 \leq x_1, x_4 \leq 2$, and $0.1 \leq x_2, x_3 \leq 10$).

Supplementary

The position update video for f_2 and f_3 is provided in the Supplementary.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.eswa.2020.113340.

References

Ashlock, D. (2006). *Evolutionary computation for modeling and optimization*. Springer Science & Business Media.

Banks, A., Vincent, J., & Anyakoha, C. (2007). A review of particle swarm optimization. part i: background and development. *Natural Computing, 6*(4), 467–484.

Castillo, O., Soto, C., & Valdez, F. (2018). A review of fuzzy and mathematic methods for dynamic parameter adaptation in the firefly algorithm. In *Advances in data analysis with computational intelligence methods* (pp. 311–321). Springer.

Chen, Y.-S., Johansson, P. J. M., Hsu, C.-C., Liao, P.-K., & Lin, S.-J. (2016). Method of maintaining multiple timing advance. US Patent 9,467,959.

Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization tasks. *Computers in Industry, 41*(2), 113–127.

Fister, I., Fister Jr, I., Yang, X.-S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation, 13*, 34–46.

Fleury, C., & Braibant, V. (1986). Structural optimization: a new dual method using mixed variables. *International journal for numerical methods in engineering, 23*(3), 409–428.

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers, 29*(1), 17–35.

Gazi, V., & Passino, K. M. (2004). Stability analysis of social foraging swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34*(1), 539–557.

Hoang, D. T. (2008). *Metaheuristics for NP-hard combinatorial optimization problems* Ph.D. thesis..

Horng, M.-H. (2012). Vector quantization using the firefly algorithm for image compression. *Expert Systems with Applications, 39*(1), 1078–1091.

Jain, L., & Katarya, R. (2019). Discover opinion leader in online social network using firefly algorithm. *Expert Systems with Applications, 122*, 1–15.

Jati, G. K., et al. (2011). Evolutionary discrete firefly algorithm for travelling salesman problem. In *Adaptive and intelligent systems* (pp. 393–403). Springer.

Kavousi-Fard, A., Samet, H., & Marzbani, F. (2014). A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting. *Expert systems with applications, 41*(13), 6047–6056.

Koziel, S., & Yang, X.-S. (2011). *Computational optimization, methods and algorithms*: 356. Springer.

Lagunes, M. L., Castillo, O., Valdez, F., Soria, J., & Melin, P. (2018). Parameter optimization for membership functions of type-2 fuzzy controllers for autonomous mobile robots using the firefly algorithm. In *North american fuzzy information processing society annual conference* (pp. 569–579). Springer.

Langari, R. K., Sardar, S., Mousavi, S. A. A., & Radfar, R. (2020). Combined fuzzy clustering and firefly algorithm for privacy preserving in social networks. *Expert Systems with Applications, 141*, 112968.

Łukasik, S., & Zak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. In *International conference on computational collective intelligence* (pp. 97–106). Springer.

Maeda, K., Fukano, Y., Yamamichi, S., Nitta, D., & Kurata, H. (2011). An integrative and practical evolutionary optimization for a complex, dynamic model of biological networks. *Bioprocess and biosystems engineering, 34*(4), 433–446.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Moses, F., & Onoda, S. (1969). Minimum weight design of structures with application to elastic grillages. *International Journal for Numerical Methods in Engineering, 1*(4), 311–331.

Nazir, M., Majid-Mirza, A., & Ali-Khan, S. (2014). Pso-ga based optimized feature selection using facial and clothing information for gender classification. *Journal of applied research and technology, 12*(1), 145–152.

Nowacki, H. (1973). Optimization in pre-contract ship design,.

Potter, M. A. (1997). *The design and analysis of a computational model of cooperative coevolution*. Citeseer Ph.D. thesis..

Potter, M. A., & De Jong, K. A. (1994). A cooperative coevolutionary approach to function optimization. In *International conference on parallel problem solving from nature* (pp. 249–257). Springer.

Rao, S. S. (2019). *Engineering optimization: theory and practice*. John Wiley & Sons.

Reynolds, A. M., & Frye, M. A. (2007). Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search. *PloS one, 2*(4), e354.

Sánchez, D., Melin, P., & Castillo, O. (2017). Optimization of modular granular neural networks using a firefly algorithm for human recognition. *Engineering Applications of Artificial Intelligence, 64*, 172–186.

Sayadi, M., Ramezanian, R., & Ghaffari-Nasab, N. (2010). A discrete firefly metaheuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations, 1*(1), 1–10.

Senthilnath, J., Omkar, S., & Mani, V. (2011). Clustering using firefly algorithm: performance study. *Swarm and Evolutionary Computation, 1*(3), 164–171.

Surjanovic, S., & Bingham, D. (2017). Virtual library of simulation experiments: Test functions and datasets. Retrieved December 3, 2018, from http://www.sfu.ca/~ssurjano.

Tamura, K., & Yasuda, K. (2011). Primary study of spiral dynamics inspired optimization. *IEEJ Transactions on Electrical and Electronic Engineering, 6*(S1), S98–S100.

Tucker, V. A. (2000). The deep fovea, sideways vision and spiral flight paths in raptors. *Journal of Experimental Biology, 203*(24), 3745–3754.

Tucker, V. A., Tucker, A. E., Akers, K., & Enderson, J. H. (2000). Curved flight paths and sideways vision in peregrine falcons (falco peregrinus). *Journal of Experimental Biology, 203*(24), 3755–3763.

Wang, G., Guo, L., Duan, H., Liu, L., Wang, H., et al. (2012). A modified firefly algorithm for ucav path planning. *International Journal of Hybrid Information Technology, 5*(3), 123–144.

Yang, X.-S. (2008). Firefly algorithm. *Nature-inspired metaheuristic algorithms, 20*, 79–90.

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169–178). Springer.

Yang, X.-S. (2010a). Firefly algorithm, levy flights and global optimization. In *Research and development in intelligent systems xxvi* (pp. 209–218). Springer.

Yang, X.-S. (2010b). Firefly algorithm, stochastic test functions and design optimisation. arXiv:1003.1409.

Yang, X.-S. (2010c). *Nature-inspired metaheuristic algorithms*. Luniver press.

Yang, X.-S., & Deb, S. (2010). Eagle strategy using lévy walk and firefly algorithms for stochastic optimization. In *Nature inspired cooperative strategies for optimization (nicso 2010)* (pp. 101–111). Springer.

Yang, X.-S., & He, X. (2013). Firefly algorithm: recent advances and applications. arXiv preprint arXiv:1308.3898.

Yang, X.-S., Hosseini, S. S. S., & Gandomi, A. H. (2012). Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Applied soft computing, 12*(3), 1180–1186.

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