Investigating the Firefly Algorithm for Continuous Space Optimization

Jinchi Li

Abstract

Inspired by the natural communication mechanism of fireflies through bioluminescent signals, the Firefly Algorithm, developed by Professor Xin-She Yang, is a bio-inspired optimization method. This report explores the mathematical principles behind the algorithm, implements simulations on benchmark peak functions, and compares its performance against the Genetic Algorithm. The results demonstrate the Firefly Algorithm's superior accuracy and applicability to continuous space optimization problems, confirming its feasibility for complex tasks such as control system parameter tuning, image segmentation, path planning, and wireless sensor networks.

1 Introduction

In nature, fireflies communicate by emitting bioluminescent signals, using brightness to attract other fireflies. Professor Xin-She Yang from the UK harnessed this behavior to develop the Firefly Algorithm (FA) [1]. FA is a swarm intelligence algorithm designed to solve complex optimization problems. This report investigates the principles behind FA, simulates its performance, and compares it with the Genetic Algorithm (GA).

2 Methodology: The Firefly Algorithm

In nature, there are approximately 2,000 species of fireflies. Most firefly species emit short, rhythmic flashes of light, though the purposes of these flashes vary across species, and the true reasons behind them are still being explored. Generally, it is believed that the biological significance of adult fireflies' bioluminescence is to attract mates by using species-specific flashing signals, thus facilitating mating and reproduction. Some fireflies use these flashing signals to hunt, while others emit light as a warning signal when stimulated. The Firefly Algorithm (FA) simulates the light-emitting behavior of fireflies as a random optimization algorithm. However, in the algorithm, some of the biological purposes of firefly bioluminescence are disregarded, focusing solely on using their brightness to search the solution space, attract other fireflies, and move towards better solutions, thereby achieving positional evolution.

In the algorithm, the attraction between fireflies is governed by two key factors: brightness and attractiveness. The brightness of a firefly depends on the objective value at its position in the search space—the higher the brightness, the better the solution at that location. The attractiveness of a firefly is related to its brightness; brighter fireflies have higher attractiveness and can draw in nearby fireflies that are dimmer. If two fireflies have equal brightness, they move randomly. Both brightness and attractiveness decrease with distance, simulating the natural absorption of light as it propagates through space.

The Firefly Algorithm is thus a bio-inspired, stochastic optimization method that models the behavior of fireflies in nature. The points in the search space represent individual fireflies, the search and optimization process mimics their attraction and movement towards each other, and the evaluation of the objective function reflects the quality of the firefly's position. The iterative process of replacing suboptimal solutions with better ones is analogous to the process of natural selection among firefly populations.

2.1 Mathematical Description

As described above, the Firefly Algorithm is characterized by two primary components: brightness and attractiveness. Brightness reflects the quality of a firefly's position and determines its movement direction, while attractiveness controls the distance a firefly moves. Through the continuous updating of brightness and attractiveness, the algorithm optimizes the objective function.

The relative brightness I of a firefly is defined as:

$$I = I_0 e^{-\gamma r^2} \tag{1}$$

where I_0 is the maximum brightness at $r = 0$, which is related to the objective function value; the better the value, the higher the brightness. γ is the light absorption coefficient, representing the natural decay of light intensity with increasing distance r, and can be set as a constant. r is the distance between two fireflies i and j.

The attractiveness β of a firefly is defined as:

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

where β_0 is the maximum attractiveness at $r = 0$, and γ is the same light absorption coefficient as in the brightness formula.

The movement of firefly i towards firefly j is determined by the following update equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha(\text{rand} - 0.5) \tag{3}$$

where x_i and x_j represent the positions of fireflies i and j, respectively; α is a step size parameter; and rand is a random factor uniformly distributed in [0, 1].

The optimization process begins by randomly scattering the fireflies in the solution space. Each firefly emits light of varying brightness depending on its location. By comparing brightness (as described by the equation above), brighter fireflies attract dimmer ones towards them, and the distance moved is determined by the degree of attractiveness. To expand the search area and avoid premature convergence to local optima, a disturbance term is added to the position update equation. After multiple iterations, all fireflies converge at the position of the brightest firefly, thereby achieving optimization.

2.2 Optimization Procedure

The Firefly Algorithm (FA) follows a bio-inspired approach based on the natural behavior of fireflies emitting light to attract others. The algorithm's detailed procedure is described as follows:

Algorithm 1 Firefly Algorithm

Initialize algorithm parameters: number of fireflies n, maximum attractiveness β_0, light absorption coefficient γ, step size α, maximum iterations $MaxGeneration$, or desired accuracy ϵ.
Randomly initialize the positions of the fireflies and calculate the objective function value for each firefly as its brightness I_i.
while stopping criteria not met **do**
    for each firefly i **do**
        for each firefly j **do**
            if $I_j > I_i$ **then**
                Calculate relative brightness I_{ij} and attractiveness β_{ij} using equations (5) and (6).
                Update the position of firefly i using equation (7).
                Apply random perturbation to the best solution.
            end if
        end for
    end for
    Recalculate brightness for each firefly based on the updated positions.
    Increment iteration counter.
end while
Output the global best solution and the corresponding objective value.

1. **Initialization**: The algorithm begins by setting the basic parameters, including the number of fireflies n, the maximum attractiveness β_0, the light absorption coefficient γ, the step size α, and either the maximum number of iterations $MaxGeneration$ or the desired accuracy ϵ. The positions of the fireflies are randomly initialized in the solution space, and the brightness of each firefly is calculated based on the objective function value at its position.

2. **Calculation of Brightness and Attractiveness**: The relative brightness I_{ij} and attractiveness β_{ij} between fireflies are calculated using Equations (5) and (6). The brightness of a firefly depends on the objective value of its position, with brighter fireflies representing better solutions. The attractiveness between fireflies decreases with increasing distance, simulating the decay of light intensity in nature.

3. **Movement of Fireflies**: Fireflies are attracted to brighter ones and adjust their positions according to Equation (7). If a firefly is located in a position that is less optimal (lower brightness), it will move towards a brighter firefly. To enhance exploration and avoid premature convergence to local optima, a random perturbation is added to the best solution.

4. **Update of Brightness**: After the positions of the fireflies have been updated, the brightness of each firefly is recalculated based on its new position. This allows the algorithm to continually refine its search for the optimal solution.

5. **Termination Criteria**: The search process is repeated until the stopping criterion is met. The algorithm can terminate either when the desired accuracy ϵ is reached or after a maximum number of iterations $MaxGeneration$. If neither condition is met, the iteration count is incremented and the search continues.

6. **Output**: Once the stopping criterion is satisfied, the algorithm outputs the global optimal solution, represented by the position of the brightest firefly, as well as the corresponding optimal objective function value.

The overall process of the Firefly Algorithm is presented in Algorithm 1. The algorithm's time complexity is $O(n^2)$, where n is the number of fireflies. This complexity arises from the pairwise comparison of brightness and the position updates performed during each iteration. FA follows a process of repeated iterations in which fireflies move closer to brighter ones until convergence at the brightest firefly's position. This indicates the optimal solution for the given function [2]. To further explore the algorithm, I simulated FA's performance on various benchmark peak functions and compared it to the Genetic Algorithm.

3 Experimental Setup and Results

In this work, we used the Firefly Algorithm (FA) and the Genetic Algorithm (GA) to simulate and test the performance of both algorithms on the following three benchmark functions:

3.1 Benchmark Functions

$$F_1(x) = \exp(-(x_1-3)^2 - (x_2-5)^2) + \exp(-x_1^2 - x_2^2), \quad |x| \leq 10 \tag{4}$$
$$F_2(x) = \exp(-(x_1-4)^2 - (x_2-4)^2) + \exp(-(x_1-4)^2 - x_2^2) \tag{5}$$
$$+ \exp(-x_1^2 - (x_2-4)^2) + 2 \times \exp(-x_1^2 - x_2^2), \quad |x| \leq 5 \tag{6}$$
$$F_3(x) = \sum_{k=1}^{2} x_k^2 - 10\cos(2\pi x_k), \quad |x| \leq 5.12 \tag{7}$$

These three functions differ in the number of local optima they contain. Specifically:

- $F_1(x)$ has two peaks in the range $|x| \leq 10$,
- $F_2(x)$ has four peaks in the range $|x| \leq 5$,
- $F_3(x)$ has many peaks in the range $|x| \leq 5.12$.

The objective of the experiment was to compare the optimization performance of FA and GA on these multimodal functions, each containing a varying number of local optima. For each function, the algorithms were run for a fixed number of iterations, and the accuracy and convergence speed of the algorithms were evaluated.

3.2 Results and Discussion

The results presented in Tables 1, 2, and 3 highlight the comparative performance of the Firefly Algorithm (FA) and the Genetic Algorithm (GA) on three different benchmark functions.

More specifically, Table 1 shows the results for $F_1(x)$, a function with 2 peaks in a 2-dimensional space. Over 50, 100, and 200 iterations, the FA consistently outperforms GA. Even at 50 iterations, FA reaches an average best value of 0.9997, compared to

Table 1: Evaluating Firefly Algorithm (FA) and Genetic Algorithm (GA) on $F_1(x)$. ABV is short for Averaged Best Value.

Function	Dimensions	Peaks	Iterations	ABV (FA)	ABV (GA)
$F_1(x)$	2	2	50	0.9997	0.9314
$F_1(x)$	2	2	100	0.9999	0.9772
$F_1(x)$	2	2	200	0.9999	0.9998

Table 2: Evaluating Firefly Algorithm (FA) and Genetic Algorithm (GA) on $F_2(x)$. ABV is short for Averaged Best Value.

Function	Dimensions	Peaks	Iterations	ABV (FA)	ABV (GA)
$F_2(x)$	2	4	20	1.9334	1.6760
$F_2(x)$	2	4	50	1.9999	1.7937
$F_2(x)$	2	4	100	1.9999	1.9997

0.9314 for GA. As the iterations increase, both algorithms converge towards the optimal solution, but FA still maintains a slight edge in accuracy. After 200 iterations, FA achieves a value of 0.9999, while GA attains 0.9998, demonstrating that both algorithms are effective, though FA shows faster convergence. Table 2 provides the results for $F_2(x)$, which has 4 peaks in a 2-dimensional space. The FA exhibits clear superiority over GA at every iteration level. At 20 iterations, FA reaches an average best value of 1.9334, significantly higher than GA's 1.6760. As the number of iterations increases, FA reaches the global optimum much faster, stabilizing at an average value of approximately 1.9999 by 50 iterations. In contrast, GA requires 100 iterations to approach a comparable value of 1.9997. These results demonstrate that FA is more efficient in finding the optimal solution, especially for functions with more complex multimodal landscapes. Table 3 shows the performance on $F_3(x)$, a function with multiple peaks in a 2-dimensional space. The FA continues to outperform GA across all iteration levels. At 50 iterations, FA reaches an average best value of 80.6902, while GA achieves a lower value of 80.3552. As the iterations increase to 200, FA further improves to 80.6985, while GA reaches 80.6238. Although both algorithms perform well on this highly multimodal function, FA consistently converges more accurately to the optimal regions of the search space.

Overall Discussion: Across all functions and iteration levels, FA consistently achieves higher accuracy and faster convergence than GA. This performance advantage is particularly pronounced for functions with more complex multimodal structures, such as $F_2(x)$ and $F_3(x)$. FA's ability to avoid premature convergence and explore the search space more effectively enables it to reach optimal solutions faster and with greater precision. These results underscore the strength of FA in handling complex optimization problems, particularly in multimodal and higher-dimensional spaces. The results demonstrated that FA consistently outperformed GA in terms of accuracy and convergence for all tested functions. In particular, FA showed an enhanced ability to escape local optima due to its random perturbation mechanism, which helps avoid premature convergence, particularly in highly multimodal functions such as $F_3(x)$.

Table 3: Evaluating Firefly Algorithm (FA) and Genetic Algorithm (GA) on $F_3(x)$. ABV is short for Averaged Best Value.

Function	Dimensions	Peaks	Iterations	ABV (FA)	ABV (GA)
$F_3(x)$	2	Multiple	50	80.6902	80.3552
$F_3(x)$	2	Multiple	100	80.6918	80.5454
$F_3(x)$	2	Multiple	200	80.6985	80.6238

4 Related Work

In recent years, researchers have developed numerous swarm intelligence optimization algorithms by studying the social behavior of insects and animals. Notable algorithms include the Genetic Algorithm (GA), Ant Colony Algorithm (ACA), Particle Swarm Optimization (PSO), Artificial Fish-Swarm Algorithm (AFSA), and Firefly Algorithm (FA). The Genetic Algorithm (GA), introduced by Holland in 1975, simulates evolutionary processes and is widely applied to combinatorial optimization, machine learning, and control systems [3]. In 1991, Dorigo and colleagues developed the Ant Colony Algorithm (ACA) to solve combinatorial optimization problems based on ant behavior [4]. The Particle Swarm Optimization (PSO) algorithm, proposed by Kennedy and Eberhart in 1995, is used for optimization in non-linear and complex domains [5].

In 2002, Li and colleagues introduced the Artificial Fish-Swarm Algorithm (AFSA), inspired by fish foraging behavior, which has been applied to optimization and neural network tuning [6]. Glowworm Swarm Optimization (GSO), developed by Krishnanand and Ghose in 2005, is effective in multimodal function optimization [7]. Finally, the Firefly Algorithm (FA), introduced by Yang in 2009, is inspired by the bioluminescent communication of fireflies and has applications in production scheduling and path planning [8].

5 Conclusion

This report presented an investigation of the Firefly Algorithm, its mathematical foundations, and comparative performance analysis with the Genetic Algorithm. The experiments confirmed FA's feasibility and efficiency in continuous optimization problems. Its potential in various real-world applications makes it a valuable tool for further study in optimization and related fields.

References

[1] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms.* Luniver Press, 2010.

[2] ——, "Firefly algorithm, stochastic test functions and design optimisation," *International journal of bio-inspired computation*, vol. 2, no. 2, pp. 78–84, 2013.

[3] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* University of Michigan Press, 1975.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[5] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[6] X.-L. Li, M.-Z. Shao, J.-X. Qian, and Y. Zhang, “Artificial fish swarm algorithm for global optimization,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 3, pp. 1248–1257, 2002.

[7] K. Krishnanand and D. Ghose, “Glowworm swarm optimization algorithm for multi-modal function optimization and applications,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 719–731, 2009.

[8] X.-S. Yang, “Firefly algorithms for multimodal optimization,” *International symposium on stochastic algorithms*, pp. 169–178, 2009.