8 Firefly Algorithms

Nature-inspired metaheuristic algorithms, especially those based on swarm intelligence, have attracted much attention in the last 10 years. The firefly algorithm (FA) appeared about five years ago in 2008, and its literature has expanded dramatically with diverse applications. In this chapter, we first introduce the standard firefly algorithm and then briefly review the variants, together with a selection of recent publications. We also analyze the characteristics of FA and try to answer the question of why FA is so efficient.

8.1 The Firefly Algorithm

FA was first developed by Xin-She Yang in late 2007 and published in 2008 [48,49]. FA was based on the flashing patterns and behavior of fireflies. Therefore, let us start first with a description of the flashing behavior of tropical fireflies.

8.1.1 *Firefly Behavior*

The flashing light of fireflies is an amazing sight in the summer sky in the tropical and temperate regions. There are about 2000 firefly species, and most fireflies produce short, rhythmic flashes. The pattern of flashes is often unique for a particular species. The flashing light is produced by a process of bioluminescence; the true functions of such signaling systems are still being debated. However, two fundamental functions of such flashes are to attract mating partners (communication) and to attract potential prey [32]. In addition, flashing may also serve as a protective warning mechanism to remind potential predators of the bitter taste of fireflies.

The rhythmic flash, the rate of flashing, and the amount of time between flashes form part of the signal system that brings both sexes together [32]. Females respond to a male's unique pattern of flashing in the same species, whereas in some species such as *Photuris*, female fireflies can eavesdrop on the bioluminescent courtship signals and even mimic the mating flashing pattern of other species so as to lure and eat the male fireflies who may mistake the flashes as a potential suitable mate. Some tropical fireflies can even synchronize their flashes, thus forming emerging biological self-organized behavior.

We know that the light intensity at a particular distance r from the light source obeys the inverse-square law. That is to say, the light intensity I decreases as the distance r increases in terms of $I \propto 1/r^2$. Furthermore, the air absorbs light, which becomes weaker and weaker as the distance increases. These two combined factors make most

Nature-Inspired Optimization Algorithms. http://dx.doi.org/10.1016/B978-0-12-416743-8.00008-7

fireflies visable to a limit distance, usually several hundred meters at night, which is good enough for fireflies to communicate.

The flashing light can be formulated in such a way that it is associated with the objective function to be optimized, which makes it possible to formulate new optimization algorithms.

8.1.2 Standard Firefly Algorithm

Now we can idealize some of the flashing characteristics of fireflies so as to develop firefly-inspired algorithms. For simplicity in describing the standard FA, we now use the following three idealized rules:

- All fireflies are unisex, so one firefly will be attracted to other fireflies regardless of their sex.
- Attractiveness is proportional to a firefly's brightness. Thus for any two flashing fireflies, the less brighter one will move toward the brighter one. The attractiveness is proportional to the brightness, both of which decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
- The brightness of a firefly is affected or determined by the landscape of the objective function.

For a maximization problem, the brightness can simply be proportional to the value of the objective function. Other forms of brightness can be defined in a similar way to the fitness function in genetic algorithms.

Based on these three rules, the basic steps of the FA can be summarized as the pseudo code shown in Figure 8.1.

Firefly Algorithm

Objective function $f(\boldsymbol{x})$, $\quad \boldsymbol{x} = (x_1, ..., x_d)^T$.
Generate an initial population of n fireflies $\boldsymbol{x}_i$ $(i = 1, 2, ..., n)$.
Light intensity I_i at $\boldsymbol{x}_i$ is determined by $f(\boldsymbol{x}_i)$.
Define light absorption coefficient γ.
while (t <MaxGeneration),
for $i = 1 : n$ (all n fireflies)
    for $j = 1 : n$ (all n fireflies) (inner loop)
    if $(I_i < I_j)$
        Move firefly i towards j.
    end if
        Vary attractiveness with distance r via $\exp[-\gamma r^2]$.
        Evaluate new solutions and update light intensity.
    end for j
end for i
Rank the fireflies and find the current global best $\boldsymbol{g}_*$.
end while
Postprocess results and visualization.

Figure 8.1 Pseudo code of the firefly algorithm (FA).

8.1.3 Variations of Light Intensity and Attractiveness

In the firefly algorithm, there are two important issues: the variation of light intensity and formulation of the attractiveness. For simplicity, we can always assume that the attractiveness of a firefly is determined by its brightness, which in turn is associated with the encoded objective function.

In the simplest case for maximum optimization problems, the brightness I of a firefly at a particular location $\boldsymbol{x}$ can be chosen as $I(\boldsymbol{x}) \propto f(\boldsymbol{x})$. However, the attractiveness β is relative; it should be seen in the eyes of the beholder or judged by the other fireflies. Thus, it will vary with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so we should allow the attractiveness to vary with the degree of absorption.

In the simplest form, the light intensity $I(r)$ varies according to the inverse-square law,

$$I(r) = \frac{I_s}{r^2}, \tag{8.1}$$

where I_s is the intensity at the source. For a given medium with a fixed light absorption coefficient γ, the light intensity I varies with the distance r. That is,

$$I = I_0 e^{-\gamma r}, \tag{8.2}$$

where I_0 is the original light intensity at zero distance $r = 0$. To avoid the singularity at $r = 0$ in the expression I_s/r^2, the combined effect of both the inverse-square law and absorption can be approximated as the following Gaussian form:

$$I(r) = I_0 e^{-\gamma r^2}. \tag{8.3}$$

Because a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{8.4}$$

where β_0 is the attractiveness at $r = 0$. Since it is often faster to calculate $1/(1 + r^2)$ than an exponential function, this function, if necessary, can conveniently be approximated as

$$\beta = \frac{\beta_0}{1 + \gamma r^2}. \tag{8.5}$$

It may be advantageous to use this approximation in some applications. Both (8.4) and (8.5) define a characteristic distance $\Gamma = 1/\sqrt{\gamma}$ over which the attractiveness changes significantly from β_0 to $\beta_0 e^{-1}$ for Eq. (8.4) or $\beta_0/2$ for Eq. (8.5).

In the actual implementation, the attractiveness function $\beta(r)$ can be any monotonically decreasing functions such as the following generalized form:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \qquad (m \geq 1). \tag{8.6}$$

For a fixed γ, the characteristic length becomes

$$\Gamma = \gamma^{-1/m} \to 1, \qquad m \to \infty. \tag{8.7}$$

Conversely, for a given length scale Γ in an optimization problem, the parameter γ can be used as a typical initial value. That is,

$$\gamma = \frac{1}{\Gamma^m}. \tag{8.8}$$

The distance between any two fireflies i and j at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively, is the Cartesian distance

$$r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}, \tag{8.9}$$

where $x_{i,k}$ is the kth component of the spatial coordinate $\boldsymbol{x}_i$ of ith firefly. In a 2D case, we have

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{8.10}$$

The movement of a firefly i attracted to another, more attractive (brighter) firefly j is determined by

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} (\boldsymbol{x}_j^t - \boldsymbol{x}_i^t) + \alpha\, \boldsymbol{\epsilon}_i^t, \tag{8.11}$$

where the second term is due to the attraction. The third term is randomization, with α being the randomization parameter, and $\boldsymbol{\epsilon}_i$ is a vector of random numbers drawn from a Gaussian distribution or uniform distribution. For example, the simplest form is $\boldsymbol{\epsilon}_i$ can be replaced by `rand` $-1/2$, where `rand` is a random-number generator uniformly distributed in [0,1]. For most of our implementation, we can take $\beta_0 = 1$ and $\alpha \in [0, 1]$.

8.1.4 Controlling Randomization

A further improvement on the convergence of the algorithm is to vary the randomization parameter α so that it decreases gradually as the optima are approaching. For example, we can use

$$\alpha = \alpha_\infty + (\alpha_0 - \alpha_\infty) e^{-t}, \tag{8.12}$$

where $t \in [0, t_{\max}]$ is the pseudo time for simulations and $t_{\max}$ is the maximum number of generations. α_0 is the initial randomization parameter, whereas α_∞ is the final value. We can also use a similar function to the geometrical annealing schedule. That is,

$$\alpha = \alpha_0 \theta^t, \tag{8.13}$$

where $\theta \in (0, 1]$ is the randomness reduction constant. In most applications, we can use $\theta = 0.95 \sim 0.99$ and $\alpha_0 = 1$.

In addition, in the current version of the FA algorithm, we do not explicitly use the current global best $\boldsymbol{g}_*$, even though we only use it to decode the final best solutions. Our simulations indicated that the efficiency may improve if we add an extra term $\lambda \boldsymbol{\epsilon}_i (\boldsymbol{g}_* - x_i)$ to the updating formula (8.11). Here λ is a parameter similar to α and β, and $\boldsymbol{\epsilon}_i$ is a vector of random numbers. These could form important topics for further research.

It is worth pointing out that (8.11) is a random walk, biased toward the brighter fireflies. If $\beta_0 = 0$, it becomes a simple random walk. Furthermore, the randomization term can easily be extended to other distributions such as Lévy flights.

The parameter γ now characterizes the variation of the attractiveness, and its value is crucially important in determining the speed of the convergence and how the FA algorithm behaves. In theory, $\gamma \in [0, \infty)$, but in practice, $\gamma = O(1)$ is determined by the characteristic length Γ of the system to be optimized. Thus, for most applications, it typically varies from 0.001 to 1000.

8.2 Algorithm Analysis

Now let us take a close look at the firefly algorithm and analyze its key characteristics.

8.2.1 Scalings and Limiting Cases

It is worth pointing out that the distance r defined in the previous section is *not* limited to the Euclidean distance. We can define other distance r in the d-dimensional hyperspace, depending on the type of problem we're interested in. For example, for job-scheduling problems, r can be defined as the time lag or time interval. For complicated networks such as the Internet and social networks, the distance r can be defined as the combination of the degrees of local clustering and the average proximity of vertices. In fact, any measure that can effectively characterize the quantities of interest in the optimization problem can be used as the "distance" r.

The typical scale Γ should be associated with the scale concerned in our optimization problem. If Γ is the typical scale for a given optimization problem for a very large number of fireflies, i.e., $n \gg k$, where k is the number of local optima, then the initial locations of these n fireflies should distribute relatively uniformly over the entire search space. As the iterations proceed, the fireflies would converge into all the local optima (including the global ones). By comparing the best solutions among all these optima, the global optima can easily be achieved. Our recent research suggests that it is possible to prove that the firefly algorithm will approach global optima when $n \to \infty$ and $t \gg 1$. In reality, it converges very quickly, as demonstrated later in this chapter.

There are two important limiting or asymptotic cases when $\gamma \to 0$ and $\gamma \to \infty$. For $\gamma \to 0$, the attractiveness is constant $\beta = \beta_0$ and $\Gamma \to \infty$. This is equivalent to saying that the light intensity does not decrease in an idealized sky. Thus, a flashing firefly can be seen anywhere in the domain. Therefore, a single (usually global) optima can easily be reached. If we remove the inner loop for j in Figure 8.1 and replace $\boldsymbol{x}_j$ with the current global best $\boldsymbol{g}_*$, then the FA degenerates into the special case of APSO, discussed earlier in this book. Subsequently, the efficiency of this special case is the same as that of PSO.

On the other hand, the limiting case $\gamma \to \infty$ leads to $\Gamma \to 0$ and $\beta(r) \to \delta(r)$, which is the Dirac delta function, meaning that the attractiveness is almost zero in the sight of other fireflies. This is equivalent to the case where the fireflies roam randomly in a very thick foggy region randomly. No other fireflies can be seen, and each firefly roams in a completely random way, which leads to simulated annealing (SA).

Because the firefly algorithm is usually a case between these two extremes, it is possible to adjust the parameter γ and α so that it can outperform both simulated annealing and PSO. In fact, FA can find the global optima as well as the local optima simultaneously and effectively. This advantage is demonstrated in detail later in the implementation.

A further advantage of FA is that different fireflies will work almost independently. It is thus particularly suitable for parallel implementation. It is even better than genetic algorithms and PSO because fireflies aggregate more closely around each optimum. We can expect that the interactions between different subregions are minimal in parallel implementation.

8.2.2 Attraction and Diffusion

The novel idea of attraction via light intensity as an exploitation mechanism was first used by Xin-She Yang in the firefly algorithm (FA) in 2007 and 2008. In FA, the attractiveness (and light intensity) is intrinsically linked with the inverse-square law of light intensity variations and the absorption coefficient. As a result, we have a novel term $\beta_0 \exp[-\gamma r^2]$, where β_0 is the attractiveness at the distance $r = 0$, and $\gamma > 0$ is the absorption coefficient [48].

The main function of such attraction is to enable an algorithm to converge quickly because these multi-agent systems evolve, interact, and attract, leading to some self-organized behavior and attractors. As the swarming agents evolve, it is possible that their attractor states will move toward the true global optimality.

This novel attraction mechanism is the first of its kind in the literature of nature-inspired computation and computational intelligence. This mechanism also motivated and inspired others to design similar or other kinds of attraction mechanisms. Other algorithms also used inverse-square laws, derived from nature. For example, the charged system search (CSS) used Coulomb's law; the gravitational search algorithm (GSA) used Newton's law of gravitation.

Whatever the attraction mechanism, from the metaheuristic point of view the fundamental principles are the same: that is, they allow the swarming agents to interact with one another and provide a forcing term to guide the convergence of the population.

Attraction mainly provides the mechanisms only for exploitation, but, with proper randomization, it is also possible to carry out some degree of exploration. However, the exploration is better analyzed in the framework of random walks and diffusive randomization. From the Markov chain point of view, random walks and diffusion are both Markov chains. In fact, Brownian diffusion such as the dispersion of an ink drop in water is a random walk. For example, the most fundamental random walks for an agent or solution $\boldsymbol{x}_i$ can be written as the following form:

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \epsilon, \tag{8.14}$$

where t is a counter of steps. Here, ϵ is a random number drawn from a Gaussian normal distribution with a zero mean. This gives an average diffusion distance of a particle or agent that is a square root of the finite number of steps t. That is, the distance is the order of $\sqrt{Dt}$ where D is the diffusion coefficient. To be more specific, the variance of the random walks in a d-dimensional case can be written as

$$\sigma^2(t) = |v_0|^2 t^2 + (2dD)t, \tag{8.15}$$

where v_0 is the drift velocity that can be taken as zero here.

This means it is possible to cover the whole search domain if t is sufficiently large. Therefore, the steps in the Brownian motion B(t) essentially obey a Gaussian distribution with zero mean and time-dependent variance. A diffusion process can be viewed as a series of Brownian motion that obeys a Gaussian distribution. For this reason, standard diffusion is often referred to as *Gaussian diffusion*. If the motion at each step is not Gaussian, the diffusion is called *non-Gaussian diffusion*. On the other hand, random walks can take many forms. If the step lengths obey other distributions, we have to deal with more generalized random walks. A very special case is when step lengths obey the Lévy distribution. Such random walks are called *Lévy flights* or *Lévy walks*.

It is worth pointing out that the original firefly algorithm was developed to combine with Lévy flights, and good performance has been achieved [49].

8.2.3 Special Cases of FA

FA is indeed rich in many ways. First, it uses attraction to influence the behavior of a population. Because local attraction tends to be stronger than long-distance attraction, the population in FA can automatically subdivide into subgroups, depending on the modality of the problem, which enables FA to deal with multimodal, nonlinear optimization problems naturally.

Furthermore, if we look at the updating Eq. (8.11) more closely, this nonlinear equation provides much richer characteristics. First, if γ is very large, attractiveness or light intensity decreases too quickly. This means that the second term in (8.11) becomes negligible, leading to the standard SA. Second, if γ is very small (i.e., $\gamma \to 0$), the exponential factor $\exp[-\gamma r_{ij}^2] \to 1$. We have

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t) + \alpha \epsilon_i^t. \tag{8.16}$$

Here, if we further set $\alpha = 0$, the Eq. (8.16) becomes a variant of differential evolution. On the other hand, if we replace $\boldsymbol{x}_j^t$ with the current global best solution $\boldsymbol{g}^*$, we have

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0(\boldsymbol{g}^* - \boldsymbol{x}_i^t) + \alpha \epsilon_i^t, \tag{8.17}$$

which is essentially the APSO introduced by Xin-She Yang in 2008 [48].

Third, we set $\beta_0 = 0$ and let ϵ_i^t be related to $\boldsymbol{x}_i$; then (8.16) becomes a pitch adjustment variant in harmony search (HS).

Therefore, we can essentially say that DE, APSO, SA, and HS are special cases of FA. Conversely, FA can be considered a good combination of all four algorithms (DE, APSO, SA, and HS), to a certain extent. Furthermore, FA uses nonlinear updating equation, which can produce richer behavior and higher convergence than the linear updating equations used in standard PSO and DE. Consequently, it is again no surprise that FA can outperform other algorithms in many applications such as multimodal optimization, classifications, image processing, and feature selection, as we will see later in the applications.

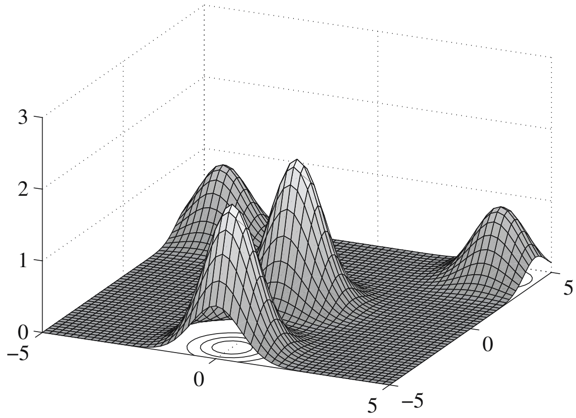

Figure 8.2 Landscape of a function with two equal global maxima.

8.3 Implementation

A demo version of FA implementation without Lévy flights can be found at the Mathworks file exchange Website.[1] In the implementation, the values of the parameters are $\alpha_0 = 0.5, \gamma = 1$ and $\beta_0 = 1$. Obviously, these parameters can be adjusted to suit solving various problems with different scales.

To demonstrate how the FA works, we use the simple example of the four-peak function

$$f(x, y) = e^{-(x-4)^2-(y-4)^2} + e^{-(x+4)^2-(y-4)^2} + 2\left[e^{-x^2-y^2} + e^{-x^2-(y+4)^2}\right],$$

where $(x, y) \in [-5, 5] \times [-5, 5]$. This function has four peaks, two local peaks with $f = 1$ at $(-4, 4)$ and (4,4) and two global peaks with $f_{\max} = 2$ at (0,0) and $(0, -4)$, as shown in Figure 8.2.

We can see that all four of these optima can be found using 25 fireflies in about 20 generations (see Figure. 8.3). So, the total number of function evaluations is about 500. This is much more efficient than most existing metaheuristic algorithms.

8.4 Variants of the Firefly Algorithm

8.4.1 FA Variants

The standard FA is very efficient, but there is still room for improvement. In the last five years, researchers have tried various ways to enhance the performance and speed up the convergence of the firefly algorithm. As a result, quite a few variants have been developed [19,42,37]. However, because the literature is expanding rapidly and more

[1] www.mathworks.com/matlabcentral/fileexchange/29693-firefly-algorithm.

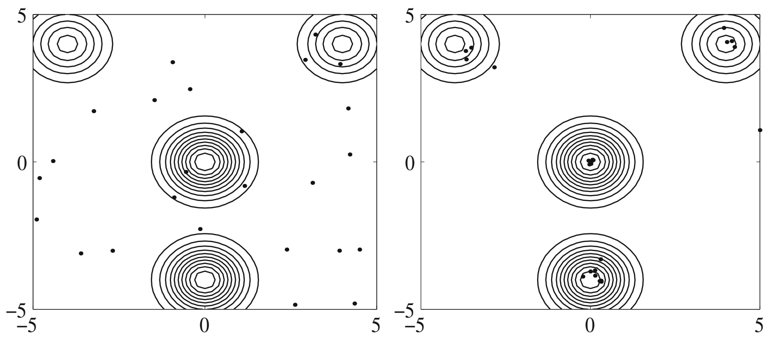

Figure 8.3 The initial locations of 25 fireflies (left) and their final locations after 20 iterations (right).

variations are appearing, it is not possible to list all these variants here. This section briefly summarizes some of these variants:

- *Discrete firefly algorithm (DFA)*. Sayadi et al. [42] extended the firefly algorithm to deal with NP-hard scheduling problems and have developed a powerful version of discrete FA. Their results show that DFA can outperform existing algorithms such as ACO. Meanwhile, Durkota independently provided a good implementation of a DFA for QAP problems [14]. In addition, an FA-based method for image segmentation, developed by Hassanzadeh et al. [24], can be far more efficient than the Otsu's method and recursive Otsu method. On the other hand, Jati and Suyanto discretized the firefly algorithm and showed its effectiveness in solving the NP-hard traveling salesman problem [29]. Furthermore, Chandrasekaran and Simon proposed an efficient binary real-coded FA variant to study the network and reliability constrained unit commitment problem [9].
- *Chaotic firefly algorithm (CFA)*. A chaotic FA was proposed by Coelho et al. in 2011, and this CFA outperformed other algorithms [11,12]. Yang studied the intrinsic chaotic characteristics of FA under different parameter ranges so that enhanced performance can be achieved by tuning β and γ [51]. At the same time, Gandomi et al. studied different chaotic maps and carried out extensive performance comparisons, concluding that some chaotic maps can indeed enhance the performance of FA by replacing some parameters in FA with these chaotic maps [21].
- *Lagrangian firefly algoirthm (LFA)*. Another interesting variant is the Lagrangian firefly algorithm, which was proposed by Rampriya et al. to solve the unit commitment problem for a power system [41].
- *Memetic firefly algorithm (MFA)*. Fister Jr. et al. developed a discrete variant of FA, called the memetic firefly algorithm, for solving combinatorial graph-coloring problems, with promising results [18].
- *Multiobjective discrete firefly algorithm (MDFA)*. Apostolopoulos and Vlacho extended FA and developed a discrete version for multi-objective optimization of the economic emission load dispatch problem, and their comparison showed that this variant is very efficient [6]. Meanwhile, Arsuaga-Rios and Vega-Rodriguez

independently proposed another multi-objective FA (MOFA) for an optimal workload management tool for minimizing energy consumption in grid computing [4]. In addition, Li and Ye used FA to solve multi-objective production scheduling systems [33]. Furthermore, Marichelvam et al. presented a discrete FA variant for the multi-objective hybrid flowshop scheduling problem [35].

- *Mulitobjective firefly algorithm (MOFA)*. Yang also extended FA for single-objective optimization to multi-objective optimization of continuous design problems [53].
- *Multi-objective enhanced firefly algorithm (MOEFA)*. Amiri et al. presented a multi-objective enhanced FA for community detection in complex networks [3].
- *Hybrid firefly algorithms (HFA)*. There are many hybrid algorithms achieved by hybridizing FA with other algorithms. For example, Giannakouris et al. combined FA with ant colony optimization and obtained good results [22]. Abdullah et al. combined FA with differential evolution to estimate nonlinear biological model parameters, and their study showed that this hybrid can be a powerful tool with less computation time for many applications [2]. An improved FA with adaptive strategies has been presented by Yan et al. [55].
- *Parallel firefly algorithm with predation (pFAP)*. Luz et al. proposed a parallel implementation of FA with predation and applied it to the inverse heat conduction problem [34].

For discrete problems and combinatorial optimization, discrete versions of FA have been developed that have superior performance [42,29,18,14], which can be used for traveling-salesman problems, graph coloring, and other applications. In addition, the extension of FA to multi-objective optimization has also been investigated [6,53]. A few studies show that chaos can enhance FA's performance [11,51], whereas other studies have attempted to hybridize FA with other algorithms to enhance their performance [22,26,27,41].

Sometimes some improvements can be carried out by slightly modifying the standard FA, but it might not be proper to classify the improved algorithm as a new variant. For example, Farahani et al. used a Gaussian distribution to replace the uniform distribution for the scaling factor α, which indeed showed some improvement [17]. On the other hand, Wang et al. introduce a discovery rate to FA and presented a modified FA for UCAV path planning [47], and their results showed the modified FA indeed performed better.

8.4.2 How Can We Discretize FA?

The standard FA was initially developed for continuous, multimodal optimization problems. It can also be applied to solve combinatorial optimization problems; however, we have to find ways to convert continuous variables into discrete variables. There are different ways to achieve this task, and one of the widely used is the so-called sigmoidal logistic function

$$S(x) = \frac{1}{1+\exp(-x)}, \tag{8.18}$$

which converts a continuous variable into a binary variable S. This S-shaped function for x gives $S \to 1$ as $x \to +\infty$, and $S \to 0$ as $x \to -\infty$. However, in practice this is not so easy to achieve, because values might not vary in the right range. A common way is to combine with a random number r with a conditional switch. That is,

$$(u > r) \to 1, \quad (u \le r) \to 0. \tag{8.19}$$

Obviously, once we have $S \in \{0, 1\}$, we can change it to $y = 2S - 1 \in \{+1, -1\}$ when necessary.

An interesting property of S is that its derivative can be calculated easily by

$$\frac{dS}{dx} = S(1 - S). \tag{8.20}$$

Other ways include signs and randomization. For example, we can use $y = \text{sign}(x)$ to generate $+1, 0, -1$. In addition, a simple (but not necessarily efficient) way is to use $y = \lfloor x \rfloor$, which is the largest integer not greater than x. Another way is to use the mod function. For example, we can use

$$x \leftarrow \lfloor x + k \rfloor \mod m, \tag{8.21}$$

where k and $m > 0$ are integers.

Another related issue is how new solutions can be generated. Once we have $S = +1$ or $S = -1$, we can use them as a step size and do local random walks at different time/iteration t. That is,

$$y^{t+1} = y^t + S. \tag{8.22}$$

However, if the design variables are binary only, we have to normalize the new variables. One way is to use mutation-like operations by swapping randomly between 0 and 1.

For randomization, one way is to use a random number associated with a discrete set. If the search space is discrete and a design variable z only takes values on a finite set, a random number u, often uniformly distributed, can be generated and map to the cardinality of the set. Once a random number is drawn, the corresponding value of the finite set can be used.

On the other hand, two key issues are how to define the distance and neighborhood in the discrete FA. For many combinatorial problems such as scheduling, the distance is not a physical distance. Therefore, care should be taken to define a distance measure properly. Any sensible metric, such as time delay, time difference, Hamming distance, edit distance, or Jaccard similarity, can be used as the distance.

For neighborhood, it is an even more challenging issue in combinatorial optimization. For example, to solve the traveling salesman problem, a neighborhood solution can be a local solution generated by swapping two links between four cities, the so-called 2-opt move [38].

It is worth pointing out that the ways of defining neighborhood solutions can significantly affect the overall performance of any implementation or variant of any nature-inspired algorithms. This is one of reasons that discrete metaheuristic algorithms form a significant part of current research activities.

8.5 Firefly Algorithms in Applications

FA has attracted much attention and has been applied to many applications [6,10,24,42,50,25,26]. Since the first original publications by Yang in 2008, more than 810 publications about FA have been published. As we can see, the literature has expanded significantly, so it is not possible to review all 800 papers here. Therefore, we sample only a fraction of these publications. Our choices might be biased, though we intended to choose these representatives in an unbiased way.

Horng et al. demonstrated that the firefly-based algorithm used the least computation time for digital image compression [25,26], whereas Zhang and Wu used FA to study image registration [58]. Banati and Bajaj used FA for feature selection and showed that FA produced consistent and better performance in terms of time and optimality than other algorithms [7].

In engineering design problems, Gandomi et al. [20] and Azad and Azad [5] confirmed that FA can efficiently solve highly nonlinear, multimodal design problems. Basu and Mahanti [8] as well as Chatterjee et al. applied FA in antenna design optimization and showed that FA can outperform ABC [10]. In addition, Zaman and Matin also found that FA can outperform PSO and obtained global best results [57]. Furthermore, FA has been use to generate alternatives for decision makers with diverse options [28].

Sayadi et al. developed a discrete version of FA that can efficiently solve NP-hard scheduling problems [42], while a detailed analysis has demonstrated the efficiency of FA over a wide range of test problems, including multi-objective load dispatch problems [6,49,52]. For example, Yang et al. solved the non-convex economic dispatch problem with valve-loading effect using FA and achieved the best results over other methods [54]. Similarly, Swarnkar solved economic load dispatch problems with reduced power losses using FA [45].

Furthermore, FA can solve scheduling and traveling salesman problems in a promising way [39,29,56]. Jati and Suyanto solved the well-known traveling salesman problem by discrete FA, whereas Yousif et al. solved scheduling jobs on grid computing using FA [56]. Both studies showed that FA is very efficient.

For queueing systems, FA has been found very efficient, as demonstrated in the detailed study by Kwiecień and Filipowicz [31]. In addition, for mixed integer programming and load dispatch problems, FA has also been found very efficient as well [9,54].

Classifications and clustering are another important area of applications of FA with excellent performance [43,40]. For example, Senthilnath et al. provided an extensive performance study by comparing FA with 11 different algorithms and concluded that FA can be efficiently used for clustering [43]. In most cases, FA outperforms all other 11 algorithms. Tang et al. provided a comprehensive review of nature-inspired algorithms for clustering [46]. In addition, FA has been applied to training neural networks [36].

For optimization in dynamic environments, FA can also be very efficient, as shown by Farahani et al. [15,16] and Abshouri et al. [1].

On the other hand, Dutta et al. showed that FA can solve isospectral spring-mass systems effectively [13]. Kazem et al. presented support vector regression with the chaos-based FA for stock market price forecasting [30]. Grewal et al. presented a study of antenna failure correction using FA and showed that FA is very flexible and effective [23].

In the context of software testing, Srivastava et al. showed that FA can be modified to generate independent test sequences efficiently and achieves superior performance [44].

8.6 Why the Firefly Algorithm is Efficient

As the literature about FA expands and new variants emerge, all point out that the FA can outperform many other algorithms. Now we may ask naturally, “Why is it so efficient?” To answer this question, let us briefly analyze FA from a different angle.

FA is swarm-intelligence-based, so it has similar advantages as other swarm-intelligence-based algorithms. However, FA has two major advantages over other algorithms: automatical subdivision and the ability to deal with multimodality. First, FA is based on attraction and attractiveness. This leads to the fact that the whole population can automatically subdivide into subgroups, and each group can swarm around each mode or local optimum. Among all these modes, the best global solution can be found. Second, this subdivision allows the fireflies to be able to find all optima simultaneously if the population size is sufficiently higher than the number of modes. Mathematically, $1/\sqrt{\gamma}$ controls the average distance of a group of fireflies that can be seen by adjacent groups. Therefore, a whole population can subdivide into subgroups with a given average distance. In the extreme case when $\gamma = 0$, the whole population will not subdivide.

This automatic subdivision ability makes FA particularly suitable for highly nonlinear, multimodal optimisation problems. In addition, the parameters in FA can be tuned to control the randomness as iterations proceed, so convergence can also be sped up by tuning these parameters. These advantages make FA flexible to deal with continuous problems, clustering and classifications, and combinatorial optimization as well.

For example, let use two functions to demonstrate the computational cost saved by FA. For details, see the more extensive studies by Yang [49]. For De Jong’s function with $d = 256$ dimensions,

$$f(\boldsymbol{x}) = \sum_{i=1}^{256} x_i^2. \tag{8.23}$$

Genetic algorithms required 25412 ± 1237 evaluations to get an accuracy of 10^{-5} of the optimal solution, whereas PSO needed 17040 ± 1123 evaluations. For FA, we achieved the same accuracy with 5657 ± 730 function evaluations. This saves computational costs of about 78% and 67% compared to GA and PSO, respectively.

For Yang’s forest function

$$f(\boldsymbol{x}) = \left(\sum_{i=1}^{d} |x_i| \right) \exp\left[-\sum_{i=1}^{d} \sin(x_i^2) \right], \quad -2\pi \le x_i \le 2\pi, \tag{8.24}$$

GA required 37079 ± 8920 with a success rate of 88% for $d = 16$, and PSO required 19725 ± 3204 with a success rate of 98%. FA obtained a 100% success rate with just

5152 ± 2493. Compared with GA and PSO, FA saved about 86% and 74%, respectively, of overall computational efforts.

In summary, FA has three distinct advantages:

- Automatic subdivision of the whole population into subgroups
- The natural capability of dealing with multimodal optimization
- High ergodicity and diversity in the solutions

All these advantages make FA unique and very efficient.

References

[1] Abshouri AA, Meybodi MR, Bakhtiary A. New firefly algorithm based on multiswarm and learning automata in dynamic environments. In: Third international conference on signal processing systems (ICSPS 2011), August 27–28, Yantai, China; 2011. p. 73–7.

[2] Abdullah A, Deris S, Anwar S, Arjunan SNV. An evolutionary firefly algorithm for the estimation of nonlinear biological model parameters. PLoS One 2013;8(3):e56310.

[3] Amiri B, Hossain L, Crawford JW, Wigand RT. Community detection in complex networks: Multi-objective enhanced firefly algorithm. Knowl Based Syst 2013;46(1):1–1.

[4] Arsuaga-Rios M, Vega-Rodriguez MA. Multi-objective firefly algorithm for energy optimization in grid environments. Swarm Intelligence. Lect Notes Comput Sci 2012;7461:350–1.

[5] Azad SK, Azad SK. Optimum design of structures using an improved firefly algorithm. Int J Optim Civil Eng 2011;1(2):327–40.

[6] Apostolopoulos T, Vlachos A. Application of the firefly algorithm for solving the economic emissions load dispatch problem. Int J Combin 2011[Article ID 523806].

[7] Banati H, Bajaj M. Firefly-based feature selection approach. Int J Comput Sci 2011;8(2):473–80.

[8] Basu B, Mahanti GK. Firefly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna. Prog Electromagnet Res B 2011;32(1):169–90.

[9] Chandrasekaran K, Simon SP. Network and reliability constrained unit commitment problem using binary real coded firefly algorithm. Int J Electr Power Energy Syst 2012;42(1):921–32.

[10] Chatterjee A, Mahanti GK, Chatterjee A. Design of a fully digital controlled reconfigurable switched beam conconcentric ring array antenna using firefly and particle swarm optimization algorithm. Prog Elelectromagnet Res B 2012;36(1):113–31.

[11] Coelho LS, de Andrade Bernert DL, Mariani VC. A chaotic firefly algorithm applied to reliability-redundancy optimization. In: 2011 IEEE Congress on Evolutionary Computation (CEC'11); 2011. p. 517–21.

[12] Coelho LS, Mariani VC. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. Comput Math Appl 2012;64(8):2371–82.

[13] Dutta R, Ganguli R, Mani V. Exploring isospectral spring-mass systems with firefly algorithm. Proc R Soc A 2011;467(2135):3222–40.

[14] Durkota K. Implementation of a discrete firefly algorithm for the QAP problem within the sage framework, B.Sc. Thesis. Czech Technical University; 2011. <http://cyber.felk.cvut.cz/research/theses/papers/189.pdf>; 2013 [accessed 23.05.13].

[15] Farahani SM, Abshouri AA, Nasiri B, Meybodi MR. Some hybrid models to improve firefly algorithm performance. Int J Artif Intell 2012;8(S12):97–117.

[16] Farahani SM, Nasiri B, Meybodi MR. A multiswarm based firefly algorithm in dynamic environments. In: Third international conference on signal processing systems (ICSPS 2011), August 27–28, Yantai, China; 2011. p. 68–72.
[17] Farahani SM, Abshouri AA, Basiri B, Meybodi MR. A Gaussian firefly algorithm. Int J Mach Learn Comput 2011;1(5):448–53.
[18] Fister I Jr, Fister I, Brest J, Yang XS. Memetic firefly algorithm for combinatorial optimization. In: Filipič B, Šilc J, editors. Bioinspired optimization methods and their applications (BIOMA2012), 24–25 May 2012, Bohinj, Slovenia; 2012. p. 75–86.
[19] Fister I, Fister I Jr., Yang XS, Brest J. A comprehensive review of firefly algorithms. Swarm Evol Comput 2013;13(1):34–46. www.sciencedirect.com/science/article/pii/S2210650213000461.
[20] Gandomi AH, Yang XS, Alavi AH. Mixed variable structural optimization using firefly algorithm. Comput Struct 2011;89(23–24):2325–36.
[21] Gandomi AH, Yang XS, Talatahari S, Alavi AH. Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simulat 2013;18(1):89–98.
[22] Giannakouris G, Vassiliadis V, Dounias G. Experimental study on a hybrid nature-inspired algorithm for financial portfolio optimization. In: SETN 2010, Lecture notes in artificial intelligence (LNAI 6040); 2010. p. 101–111.
[23] Grewal NS, Rattan M, Patterh MS. A linear antenna array failure correction using firefly algorithm. Prog Electromagnet Res 2012;27:241–54.
[24] Hassanzadeh T, Vojodi H, Moghadam AME. An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm. In: Proceedings of the seventh international conference on natural computation (ICNC 2011); 2011. p. 1817–21.
[25] Horng MH, Lee YX, Lee MC, Liou RJ. Firefly metaheuristic algorithm for training the radial basis function network for data classification and disease diagnosis. In: Parpinelli R, Lopes HS, editors. Theory and new applications of swarm intelligence; 2012. p. 115–32.
[26] Horng MH. Vector quantization using the firefly algorithm for image compression. Expert Syst Appl 2012;39(2):1078–91.
[27] Horng MH, Liou RJ. Multilevel minimum cross-entropy threshold selection based on the firefly algorithm. Expert Syst Appl 2011;38(9):14805–11.
[28] Imanirad R, Yang XS, Yeomans JS. Modeling-to-generate-alternatives via the firefly algorithm. J Appl Oper Res 2013;5(1):14–21.
[29] Jati GK, Suyanto S. Evolutionary discrete firefly algorithm for traveling salesman problem. ICAIS 2011, Lecture notes in artificial intelligence (LNAI 6943) 2011; 2011:393–403.
[30] Kazem A, Sharifi E, Hussain FK, Saberi M, Khadeer O. Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Appl Soft Comput 2013;13(2):947–58.
[31] Kwiecień J, Filipowicz B. Firefly algorithm in optimization of queueing systems. Bull Pol Acad Sci Tech Sci 2012;60(2):363–8.
[32] Lewis SM, Cratsley CK. Flash signal evolution, mate choice and predation in fireflies. Ann Rev Entomol 2008;53(2):293–321.
[33] Li HM, Ye CM. Firefly algorithm on multi-objective optimization of production scheduling system. Adv Mech Eng Appl 2012;3(1):258–62.
[34] Luz EFP, Campos Velho HF, Becceneri JC. Firefly algorithm with predation: a parallel implementation applied to inverse heat conduction problem. In: Proceedings of 10th world congress on computational mechanics (WCCM 2012; conference presentation); 2012.
[35] Marichelvam MK, Prabaharan T, Yang XS. A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. IEEE Transl Evol Comput 2013. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6412790.

[36] Nandy S, Sarkar PP, Das A. Analysis of nature-inspired firefly algorithm-based back-propagation neural network training. Int J Comput Appl 2012;43(22):8–16.
[37] Nasiri B, Meybodi MR. Speciation-based firefly algorithm for optimization in dynamic environments. Int J Artif Intell 2012;8(S12):118–32.
[38] Ouaarab A, Ahiod B, Yang XS. Discrete cuckoo search algorithm for the travelling salesman problem. Neural computing and applications 2013. http://link.springer.com/article/10.1007%2Fs00521-013-1402-2.
[39] Palit S, Sinha S, Molla M, Khanra A, Kule M. A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm. In: Second international conference on computer and communication technology (ICCCT), 15–17 September 2011, India; 2011. p. 428–32.
[40] Rajini A, David VK. A hybrid metaheuristic algorithm for classification using micro array data. Int J Sci Eng Res 2012;3(2):1–9.
[41] Rampriya B, Mahadevan K, Kannan S. Unit commitment in deregulated power system using Lagrangian firefly algorithm. In: Proceedings of IEEE international conference on communication control and computing technologies (ICCCCT2010); 2010. p. 389–93.
[42] Sayadi MK, Ramezanian R, Ghaffari-Nasab N. A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. Int J Ind Eng Comput 2010;1(1):1–10.
[43] Senthilnath J, Omkar SN, Mani V. Clustering using firefly algorithm: performance study. Swarm Evol Comput 2011;1(3):164–71.
[44] Srivastava PR, Mallikarjun B, Yang XS. Optimal test sequence generation using firefly algorithm. Swarm Evol Comput 2013;8(1):44–53.
[45] Swarnkar KK. Economic load dispatch problem with reduce power losses using firefly algorithm. J Adv Comput Sci Technol 2012;1(2):42–56.
[46] Tang R, Fong S, Yang XS, Deb S. Integrating nature-inspired optimization algorithm to K-means clustering. In: The seventh international conference on digital information management (ICDIM2012), 22–24 August. Macau: IEEE Publications; 2012. p. 116–23.
[47] Wang GG, Guo LH, Duan H, Liu L, Wang HQ. A modified firefly algorithm for UCAV path planning. Int J Hybrid Inform Technol 2012;5(3):123–44.
[48] Yang XS. Nature-inspired metaheuristic algorithms. 1st ed. Frome, UK: Luniver Press; 2008.
[49] Yang XS. Firefly algorithms for multimodal optimisation. In: Watanabe O, Zeugmann T, editors. Proceedings fifth symposium on stochastic algorithms, foundations and applications. Lecture notes in computer science, vol. 5792; 2009. p. 169–78.
[50] Yang XS. Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2010;2(2):78–84.
[51] Yang XS. Chaos-enhanced firefly algorithm with automatic parameter tuning. Int J Swarm Intell Res 2012;2(4):125–36.
[52] Yang XS. Swarm-based metaheuristic algorithms and no-free-lunch theorems. In: Parpinelli R, Lopes HS, editors. Theory and new applications of swarm intelligence. Intech Open Science; 2012. p. 1–16.
[53] Yang XS. Multiobjective firefly algorithm for continuous optimization. Eng Comput 2013;29(2):175–84.
[54] Yang XS, Hosseini SSS, Gandomi AH. Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl Soft Comput 2012;12(3):1180–6.
[55] Yan X, Zhu YL, Wu JW, Chen H. An improved firefly algorithm with adaptive strategies. Adv Sci Lett 2012;16(1):249–54.
[56] Yousif A, Abdullah AH, Nor SM, Abdelaziz A. Scheduling jobs on grid computing using firefly algorithm. J Theor Appl Inform Technol 2011;33(2):155–64.

[57] Zaman MA, Matin MA. Nonuniformly spaced linear antenna array design using firefly algorithm. Int J Microw Sci Technol 2012;vol. 2012 [Article ID: 256759].

[58] Zhang YD, Wu LN. A novel method for rigid image registration based on firefly algorithm. Int J Res Rev Soft Intell Comput 2012;2(2):141–6.