The Firefly Algorithm: An Introduction

Presentation · April 2020

1 author:

Xin-She Yang
Middlesex University, UK

The Firefly Algorithm: An Introduction

Xin-She Yang

Middlesex University London

For details, please read my book:

Nature-Inspired Optimization Algorithms, Elsevier, (2014).

Matlab codes are downloadable from
https://uk.mathworks.com/matlabcentral/profile/authors/3659939-xs-yang

Almost Everything is Optimization

Almost everything is optimization ... or needs optimization ...

- Maximize efficiency, accuracy, profit, performance, sustainability, ...
- Minimize costs, wastage, energy consumption, travel distance/time, CO_2 emission, impact on environment, ...

Mathematical Optimization

Objectives: maximize or minimize $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})]$,

$$\boldsymbol{x} = (x_1, x_2, ..., x_D) \in \mathbb{R}^D,$$

subject to multiple equality and/or inequality design constraints:

$$h_i(\boldsymbol{x}) = 0, \quad (i = 1, 2, ..., M),$$

$$g_j(\boldsymbol{x}) \leq 0, \quad (j = 1, 2, ..., N).$$

In case of $m = 1$, it becomes a single-objective optimization problem.

Optimization problems can usually be very difficult to solve, especially large-scale, nonlinear, multimodal problems.

In general, we can solve only 3 types of optimization problems:

- Linear programming
- Convex optimization
- Problems that can be converted into the above two

Everything else seems difficult, especially for large-scale problems.
For example, combinatorial problems tend to be really hard – NP-hard!

Deep Learning

The objective in deep nets may be convex, but the domain is not convex and it's a high-dimensional problem.

$$\text{Minimize} \;\; E(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} \Big[u_i(\boldsymbol{x}_i, \boldsymbol{w}) - \bar{y}_i \Big]^2,$$

subject to various constraints.

Optimization Techniques

There are a wide spectrum of optimization techniques and tools.

Traditional techniques

- Linear programming (LP) and mixed integer programming.
- Convex optimization and quadratic programming.
- Nonlinear programming: Newton's method, trust-region method, interior point method, ..., barrier Method, ... etc.

But most real-world problems are not linear or convex, thus traditional techniques often struggle to cope, or simply do not work...

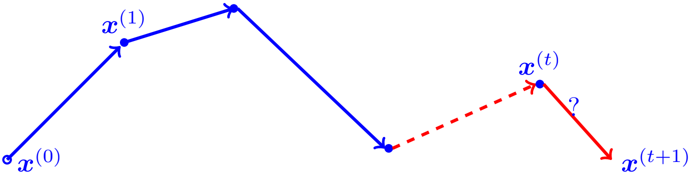
New Trends – Nature-Inspired Metaheuristic Approaches

- Evolutionary algorithms (evolutionary strategy, genetic algorithms)
- Swarm intelligence (e.g., ant colony optimization, particle swarm optimization, firefly algorithm, cuckoo search, ...)
- Stochastic, population-based, nature-inspired optimization algorithms

The Essence of an Algorithm

Essence of an Optimization Algorithm

To generate a better solution point $\boldsymbol{x}^{(t+1)}$ (a solution vector) from an existing solution $\boldsymbol{x}^{(t)}$. That is, $\boldsymbol{x}^{(t+1)} = A(\boldsymbol{x}^{(t)}, \alpha)$ where α is a set of algorithm-dependent parameters.



Population-based algorithms use multiple, interacting paths.

Different algorithms

Different ways for generating new solutions!

Main Problems with Traditional Algorithms

What's Wrong with Traditional Algorithms?

- Traditional algorithms are mostly local search, thus they cannot guarantee global optimality (except for linear and convex optimization).
- Results often depend on the initial starting points (except linear and convex problems). Methods tend to be problem-specific (e.g., k-opt, branch and bound).
- Struggle to cope problems with discontinuity.

Nature-Inspired Optimization Algorithms

Heuristic or metaheuristic algorithms (such as the firefly algorithm) tend to be a global optimizer so as to

- Increase the probability of finding the global optimality (as a global optimizer)
- Solve a wider class of problems (treating them as a black-box)
- Draw inspiration from nature (e.g., swarm intelligence)

But they can be potentially more computationally expensive.

Firefly Algorithm

The firefly algorithm (FA) was developed by Xin-She Yang in 2008.

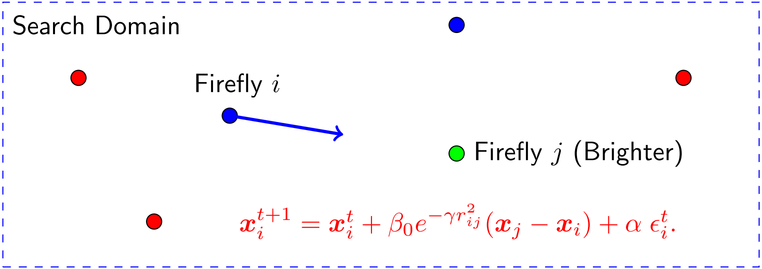
Firefly Video (YouTube)

Fireflies in Nature

- There are about 2000 firefly species, and most fireflies produce short, rhythmic flashes by bioluminescence.
- Flashing is the signaling system for fireflies: to attract mating partners (communications) and to attract potential prey (hunting), though the true functions of such flashes are still being debated.
- Flashing rhythm and timing can vary from species to species. Synchronization can occur, leading to self-organized behaviour.
- Light can be absorbed and thus brightness varies.

Firefly Behaviour and Idealization (Yang, 2008)

- Fireflies are unisex and brightness varies with distance.
- Less bright ones will be attracted to brighter ones.
- If no brighter firefly can be seen, a firefly will move randomly.



Here, $\boldsymbol{x}_i$ is the solution vector (or position of firefly i) in the search space at iteration t. β_0 is the attractiveness at zero distance (i.e., $r_{ij} = 0$), and γ is the absorption coefficient. The random vector ϵ_i^t should be drawn from a normal distribution, and the steps are scaled by a factor α.

Algorithmic Equation of FA

Attractiveness

The attractiveness β of a firefly is given by

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

where β_0 is the attractiveness at zero distance ($r = 0$).

Distance

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

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

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

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

FA Pseudocode

Algorithm 1: Firefly algorithm.

Data: Objective functions $f(\boldsymbol{x})$

Result: Best or optimal solution

Initialization of parameters (n, α, β, and γ);
Generate an initial population of n fireflies $\boldsymbol{x}_i$ $(i = 1, 2, ..., n)$;
while *(t <MaxGeneration)* **do**
    for $i = 1 : n$ *(all n fireflies)* **do**
        for $j = 1 : n$ *(all n fireflies) (inner loop)* **do**
            if *($I_i < I_j$)* **then**
                Move firefly i towards j (for maximization problems);
            end
            Vary attractiveness with distance r via $\exp[-\gamma r^2]$;
            Update the solution and evaluate new solutions;
        end
    end
    Rank the fireflies and find the current global best $\boldsymbol{g}_*$;
end
Postprocess results and visualization;

Firefly Algorithm

- The objective landscape maps to a light-based landscape, and fireflies swarm into the brightest points/regions.
- There is no $g*$, therefore, there is no leader. FA as a nonlinear iterative system, the subdivision of the whole swarm into multiswarms is possible.

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} (\boldsymbol{x}_j - \boldsymbol{x}_i) + \alpha\, \epsilon_i^t, \quad r_{ij} = \left\| \boldsymbol{x}_i^t - \boldsymbol{x}_j^t \right\|.$$

The factor in the second term is the attractiveness $\beta = \beta_0 e^{-\gamma r_{ij}^2}$, whereas the third term corresponds to perturbations/random walks.

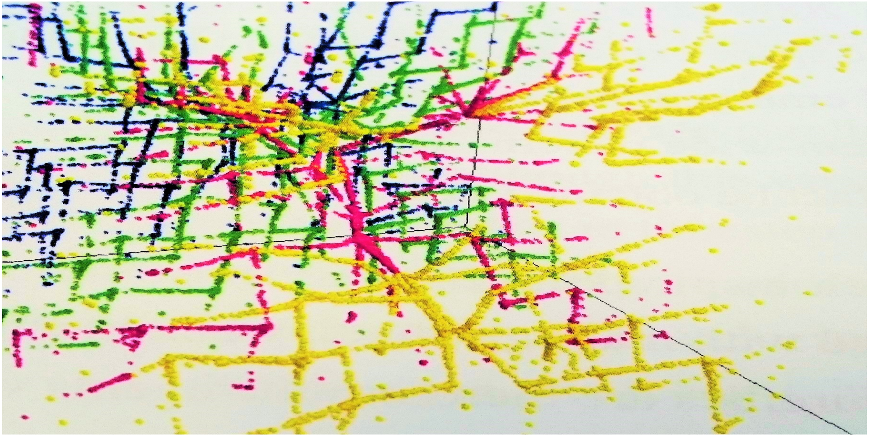
Analysis and special cases

- If $\gamma \to 0$, the attractiveness $\beta = \beta_0 e^{-\gamma r_{ij}^2} \to \beta_0$ and fireflies are visible in the whole domain. If $\gamma \to \infty$, $\beta \to \delta(r)$ (zero visibility) and fireflies move randomly (by random walks).
- Parameter α controls the strength of random walks, which should be reduced gradually during iterations.

Therefore, $\beta = O(1)$ or $\gamma = \frac{1}{L^2}$ where L is the length scale of the problem. In addition, $\alpha = \alpha_0 \theta^t$ where $0 < \theta < 1$.

FA Demo and Advantages

Fireflies can take fractal-like search paths (sparse paths, but large coverage in the search space). E.g., 3D Rosenbrock function (Husselmann, 2014).



Why is FA so efficient?

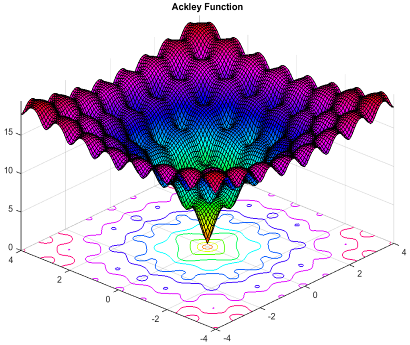
Advantages of Firefly Algorithm over PSO

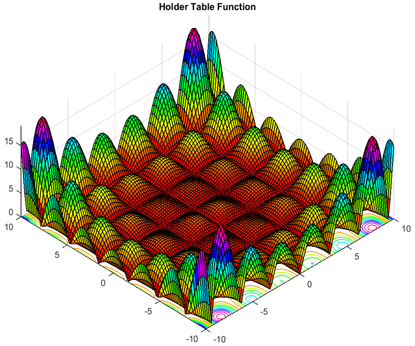
- Automatically subdivide the whole population into subgroups, and each subgroup swarms around a local mode/optimum.
- Control modes/ranges by varying γ.
- Control randomization by tuning parameters such as α.
- Suitable for multimodal, nonlinear, global optimization problems.

Typical Parameter Values

- Population size: $n = 20$ to 40 (up to 100 if necessary).
- $\beta_0 = 1$, $\gamma = 0.01$ to 10 (typically, $\gamma = 0.1$).
- $\alpha_0 = 1$, $\theta = 0.9$ to 0.99 (typically, $\theta = 0.97$).
- Number of iterations $t_{\max} = 100$ to 1000.

Subswarms and Multimodal Problems





Firefly Algorithm (Demo Video at Youtube) [Please click to start]

Firefly Algorithm is Not PSO

Main differences

- FA uses a nonlinear attraction mechanism (inverse-quare law plus exponential decay). PSO mechanism is simply linear $(x_i^t - g^*)$.
- The population in the FA can subdivide into subgroups and thus can form multi-swarms automatically (PSO cannot).
- The standard FA does not use g^* (though PSO uses g^*).
$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} (\boldsymbol{x}_j - \boldsymbol{x}_i) + \alpha \, \epsilon_i^t.$
- FA can find multiple optimal solutions simultaneously (PSO cannot).
- FA has a fractal-like search structure (PSO does not).

FA Variants for specific applications:

• Continuous optimization • Mixed integer programming • Discrete FA for combinatorial optimization such as TSP • Multiobjective FA ...
• Chaotic FA • FA for image processing, ...

Firefly Algorithm (Demo Codes) and References

FA Demo Codes

- The standard FA demo code in Matlab can be found at the Mathswork File Exchange.
  https://uk.mathworks.com/matlabcentral/fileexchange/74769-the-standard-firefly-algorithm-fa
- The multi-objective firefly algorithm (MOFA) code is also available at
  https://uk.mathworks.com/matlabcentral/fileexchange/74755-multiobjective-firefly-algorithm-mofa

Some References

- Xin-She Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, (2008).
- Xin-She Yang, Firefly algorithm, stochastic test functions and design optimisation, *Int. J. Bio-Inspired Computation*, vol. 2, no. 2, 78–84 (2010).
- Xin-She Yang, Multiobjective firefly algorithm for continuous optimization, *Engineering with Computers*, vol. 29, no. 2, 175–184 (2013).
- Xin-She Yang, Cuckoo Search and Firefly Algorithm: Theory and Applications, Springer, (2013).
- Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier Insights, (2014).