ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/340540369

Flower Pollination Algorithm: An Introduction

Presentation · April 2020

CITATIONS
0

READS
169

1 author:

Xin-She Yang
Middlesex University, UK
521 PUBLICATIONS 36,657 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Nature-Inspired Optimization Algorithms View project

Project Control system, Biomedical engineering, Signal processing View project

All content following this page was uploaded by Xin-She Yang on 10 April 2020.

The user has requested enhancement of the downloaded file.

Flower Pollination 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.

Key Components for Optimization

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...

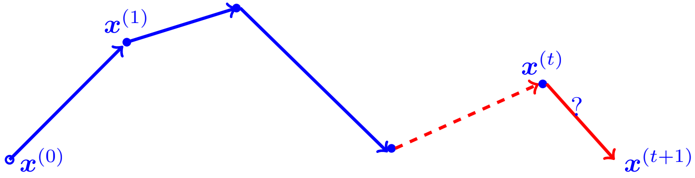
New Trends – Nature-Inspired Metaheuristic Approaches

- 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 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 (e.g., ant colony optimization, particle swarm optimization, firefly algorithm, bat algorithm, cuckoo search, differential evolution, flower pollination algorithm, etc) 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.

Flower Pollination Algorithm (FPA) [Yang, 2012]

Pollination Characteristics

Flowering plants have been evolving for at least 125 million years, and about 90% of flower pollination needs pollinators. There are about 200 000 varieties of pollinators such as insects and birds. **BBC Video (Flower Pollination) at Youtube** [click to start]

- Biotic pollination and cross pollination (about 90% of flowering plants) via pollinators (e.g., insects and animals). [Feature 1]
- Abiotic and self-pollination (local, winds) (about 10%). [Feature 2]
- Flower constancy (e.g., hummingbirds)/flower-pollinator co-evolution. [Feature 3]
- Pollinators can fly for a long distance (thus possible Lévy flights). [Feature 4]

Yang, X.S., Flower pollination algorithm for global optimization. In: Unconventional Computation and Natural Computation. Lecture Notes in Computer Science vol. 7445, 240–249, Springer, Berlin (2012).

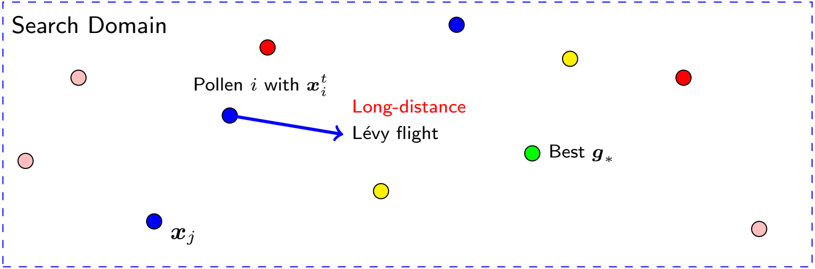
FPA Idealization (Yang, 2010)

- Feature 1 and flower constancy (feature 3) can be represented mathematically as

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \gamma L(\lambda)(\boldsymbol{g}_* - \boldsymbol{x}_i^t). \quad \text{[Direction of the search is } (\boldsymbol{g}_* - \boldsymbol{x}_i^t)\text{]}$$

- Both feature 2 and feature 3 can be represented as

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \epsilon\,(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t).$$



Here, $\boldsymbol{x}_i$ is the solution vector (or position of nest i) and $\boldsymbol{g}_*$ is the current best.

Flower Pollination Algorithm

Global pollination

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \gamma L(\lambda)\,(\boldsymbol{g}_* - \boldsymbol{x}_i^t), \tag{1}$$

where $\boldsymbol{x}$ = a solution vector. $\boldsymbol{g}_*$ = best solution (so far). Step sizes are drawn randomly from the Lévy distribution:

$$L(\lambda) \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}},$$

Local pollination

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i + \epsilon\,(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t), \tag{2}$$

where ϵ = uniformly distributed random number.

Switching Probability p

Control which branch/equation for search.

FPA Pseudocode

Algorithm 1: Flower pollination algorithm.

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

Initialize a population of n flowers/pollen gametes with random solutions;
Find the best solution $\boldsymbol{g}_*$ in the initial population;
Define a switch probability $p \in [0, 1]$;
while *($t <$MaxGeneration)* **do**
    for $i = 1 : n$ *(all n flowers in the population)* **do**
        if *rand* $< p$ **then**
            Draw a (d-dimensional) step vector L from a Lévy distribution;
            Global pollination via $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \gamma L\,(\boldsymbol{g}_* - \boldsymbol{x}_i^t)$;
        else
            Draw ϵ from a uniform distribution in [0,1];
            Do local pollination via $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \epsilon(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t)$;
        end
        Evaluate new solutions;
        If new solutions are better, update them in the population;
    end
    Find the current best solution $\boldsymbol{g}_*$;
end

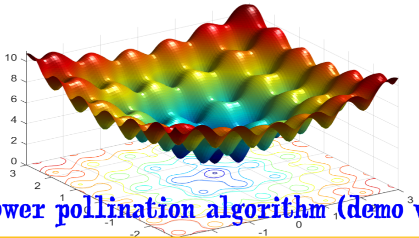
Typical Parameter Values

- Population size: $n = 20$ to 40 (up to 100 if necessary).
- Frequency: $f_{\min} = 0$, $f_{\max} = O(1)$ (typically $f_{\max} = 1$ or 2).
- Loudness: $A_0 = 1$, $\alpha = 0.9$ to 0.99 (typically $\alpha = 0.97$).
- Pulse emission rate: $r_0 = 1$, $\gamma = 0$ to 0.5 (typically $\gamma = 0.1$). Scaling: $\sigma = 0.5$.
- Number of iterations $t_{\max} = 100$ to 1000.

Demo: Ackley Function

$$f(x,y) = -20\exp[-0.2\sqrt{0.5(x^2+y^2)}] - \exp\left\{\frac{1}{2}[\cos(2\pi x) + \cos(2\pi y)]\right\} + 20 + e.$$

Optimal solution $f_{\min} = 0$ at $(0,0)$.



Flower pollination algorithm (demo video at Youtube) [Click to start]

FPA demo

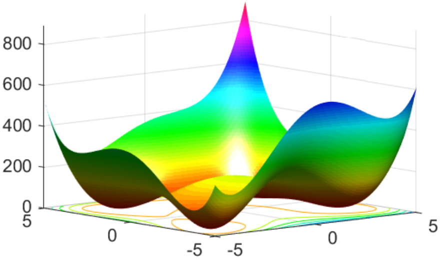
Himmelblau Function

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2, \quad (x,y) \in [-5,5]^2$$

with four global minima $f_{\min} = 0$ at

$$\boldsymbol{x}_1 = (3, 2), \qquad \boldsymbol{x}_2 = (-2.805118, 3.131312),$$
$$\boldsymbol{x}_3 = (-3.779310, -3.283186), \qquad \boldsymbol{x}_4 = (3.584428, -1.848126).$$

FPA can find all these four minima simultaneously.



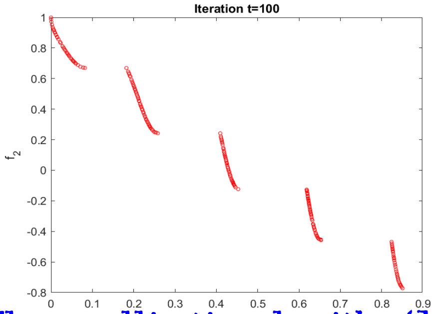
Flower pollination algorithm (demo video at Youtube) [Click to start]

Multiobjective flower pollination algorithm (MOFPA)

The Pareto front of a 30-dimensional optimization problem with two objectives

$$f_1(x) = x_1, \qquad f_2(x) = g\left[1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f_1)\right],$$

$$g(x) = 1 + \frac{9}{d-1}\sum_{i=2}^{d} x_i, \qquad x_1 \in [0,1],\ i = 2, ..., d, \quad d = 30.$$



Flower pollination algorithm (demo video at Youtube) [Click to start]

FPA (Demo Codes) and References

Flower Pollination Algorithm Demo Codes

The standard FPA demo in Matlab can be found at the Mathswork File Exchange
https://uk.mathworks.com/matlabcentral/fileexchange/74765-the-standard-flower-pollination-algorithm-fpa
The multi-objective flower pollination algorithm (MOFPA) code is also available at
https://uk.mathworks.com/matlabcentral/fileexchange/74750-multi-objective-flower-pollination-algorithm-mofpa

Some References

- Xin-She Yang, Flower pollination algorithm for global optimization, in: *Unconventional Computation and Natural Computation*, Lecture Notes in Computer Science, vol. 7445, 240–249 (2012).
- Xin-She Yang, M. Karamanoglu, X. S. He, Multi-objective flower algorithm for optimization, *Procedia Computer Science*, vol. 18, no. 1, 861–868 (2011).
- Xin-She Yang, M. Karamanoglu, X. S. He, Flower pollination algorithm: a novel approach for multiobjective optimization, *Engineering Optimization*, vol. 46, no. 9, 1222–1237 (2013).
- Xin-She Yang, Cuckoo Search and Firefly Algorithm: Theory and Applications, Springer, (2013).
- Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier Insights, (2014).

View publication stats