The Bat Algorithm: An Introduction

Presentation · April 2020

1 author:

Xin-She Yang
Middlesex University, UK

The Bat 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...

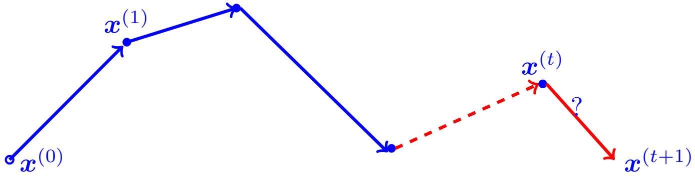
New Trends – Nature-Inspired Metaheuristic Approaches

- 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.

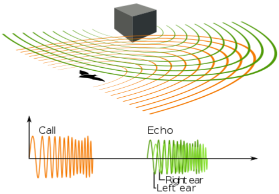
Nature-Inspired Optimization Algorithms

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.

Bat Algorithm (Yang, 2010)



BBC Video at Youtube [Click to start]

Microbats use echolocation for hunting

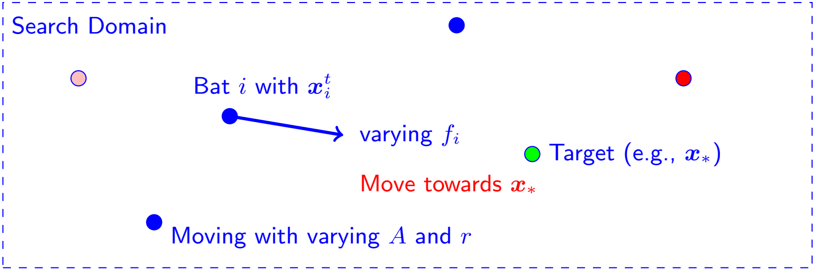
- Ultrasonic short pulses as loud as 110dB with a short period of 5 to 20 ms. Frequencies of 25 kHz to 100 kHz.
- Bats usually speed up their pulse-emission rate and increase their loudness when homing at a prey.
- Frequencies are varied/tuned so as to increase the detection resolution (by reducing wavelengths of the signals).

Xin-She Yang, A new metaheuristic bat-inspired algorithm, Nature-Inspired Cooperative Strategies for Optimization (NICSO 2010), pp. 65-74, Springer.

Sound velocity (v) = frequency (f) $\times$ wavelength (λ). So $\lambda = v/f \sim 2$ mm to 14 mm.

Echolocation of Microbats and Idealization (Yang, 2010)

- All bats use echolocation to sense distance, and they also "know" the direction of the food/prey.
- Bats fly randomly with velocity $\boldsymbol{v}_i$ at position $\boldsymbol{x}_i$. They can automatically adjust the frequency (or wavelength) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target.
- Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum value $A_{\min}$.



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

Bat Algorithm

Algorithmic equations

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \quad \beta \in [0,1] \text{ (random)},$$
$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1}, \quad \boldsymbol{v}_i^{t+1} = \boldsymbol{v}_i^t + (\boldsymbol{x}_i^t - \boldsymbol{x}_*)\, f_i.$$

If the switch condition is true (based on $\texttt{rand} > r_i$ and $\texttt{rand} < A_i$), we have

$$\boldsymbol{x}_{\text{new}} = \boldsymbol{x}_* + \sigma\, \epsilon_t\, \mathbb{A}^t$$

Here, ϵ_t is a random number in [0,1], and σ is a scaling factor.
$\boldsymbol{x}_*$=best solution found so far. $\mathbb{A}^t$ is the averaged loudness.

Variations of Loudness (A) and Pulse Rate (r)

$$A_i^{t+1} = \alpha A_i^t, \quad \alpha \in (0,1],$$
$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \quad \gamma \in (0,1).$$

Thus, $A_i^t \to 0$ and $r_i^t \to r_0$ as $t \to \infty$.

[X. S. Yang and X. He, Bat Algorithm: Literature Review and Applications, Int. J. Bio-Inspired Computation, vol.5, no. 3, 141-149 (2013).]

Bat Algorithm Pseudocode

Algorithm 1: Bat algorithm.

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

Result: Best or optimal solution

Initialize the bat population $\boldsymbol{x}_i$ and $\boldsymbol{v}_i \ \ (i = 1, 2, ..., n)$;
Initialize frequencies f_i, pulse rates r_i and the loudness A_i;
while *(t <Max number of iterations)* **do**
    Generate new solutions by adjusting frequency;
    Update velocities and locations/solutions;
    if *(rand $> r_i$)* **then**
        Select a solution among the best solutions;
        Generate a local solution around the selected best solution;
    end
    Generate a new solution by flying randomly;
    if *(rand $< A_i$ & $f(\boldsymbol{x}_i) < f(\boldsymbol{x}_*)$)* **then**
        Accept the new solutions;
        Increase r_i and reduce A_i;
    end
    Rank the bats and find the current best $\boldsymbol{x}_*$;
end

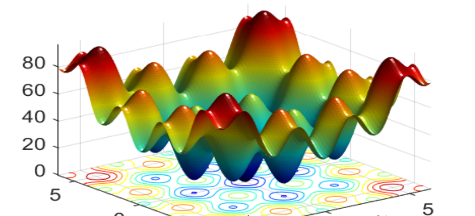
Typical Parameter Values

- 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: Eggcrate Function

$$f(x,y) = x^2 + y^2 + 25(\sin^2 x + \sin^2 y), \quad (x,y) \in [-2\pi, 2\pi]^2.$$

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

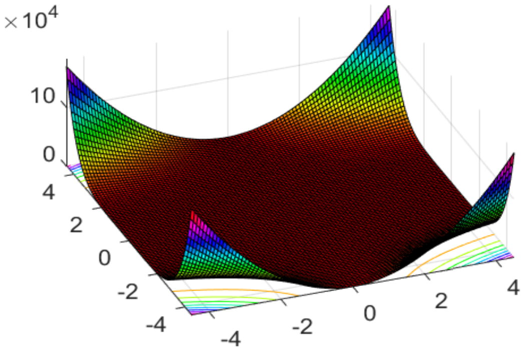


Bat Algorithm (Demo video at Youtube) [Please click to start]

Beale Function

$$f(x,y) = (1.5 - x - xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2, \quad (x,y) \in [-4.5, 4.5]^2.$$

Its landscape is relatively flat with $f_{\min} = 0$ at (3,0.5).



Bat Algorithm (Demo video at Youtube) [Please click to start]

Multi-objective Bat Algorithm (MOBA)

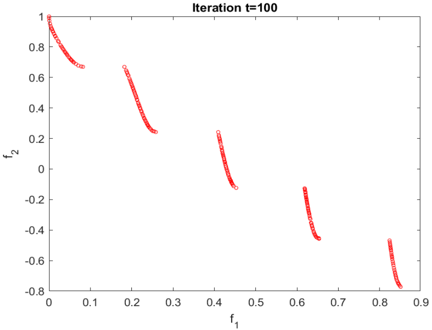
For example, the so-called ZDT function with $D = 30$ dimensions

$$\text{minimize} \quad f_1(\boldsymbol{x}) = x_1, \quad \text{and} \quad f_2(\boldsymbol{x}) = g(\boldsymbol{x})h(\boldsymbol{x}), \quad \boldsymbol{x} \in [0,1]^{30},$$

where

$$g(\boldsymbol{x}) = 1 + \frac{9}{29} \sum_{j=2}^{D=30} x_j, \quad h(\boldsymbol{x}) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} \sin(10\pi f_1),$$

has a nonconvex Pareto front in the domain $0 \leq x_i \leq 1$ where $i = 1, 2, ..., 30$.



Bat Algorithm (Demo video at Youtube) [Please click to start]

Bat Algorithm (Demo Codes) and References

Bat Algorithm Demo Codes

- The standard BA demo in Matlab can be found at the Mathswork File Exchange
  https://uk.mathworks.com/matlabcentral/fileexchange/74768-the-standard-bat-algorithm-ba
- The multi-objective bat algorithm (MOBA) code is also available at
  https://uk.mathworks.com/matlabcentral/fileexchange/74753-multiobjective-bat-algorithm-moba

Some References

- Xin-She Yang, A new metaheuristic bat-inspired algorithm, in: Nature-Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, pp. 65–74 (2010).
- Xin-She Yang, Bat algorithm for multi-objective optimisation, *Int. J. Bio-Inspired Computation*, vol. 3, no. 5, 267–274 (2011).
- Xin-She Yang, Amir Gandomi, Bat algorithm: a novel approach for global engineering optimization, *Engineering Computations*, vol. 29, no. 5, 464–483 (2012).
- Xin-She Yang, Xingshi He, Bat algorithm: literature review and applications, *Int. J. Bio-Inspired Computation*, vol. 5, no. 3, 141–149 (2013).
- Xin-She Yang, Cuckoo Search and Firefly Algorithm: Theory and Applications, Springer, (2013).
- Xin-She Yang, Nature-Inspired Optimization Algorithms, Elsevier Insights, (2014).