Artificial Bee Colony In Optimization

Alexandr Ivanov

MIPT DCAM

Abstract

- Abstract Swarm intelligence is a research branch that models the population of interacting agents or swarms that are able to self-organize. An ant colony, a flock of birds or an immune system is a typical example of a swarm system. Bees swarming around their hive is another example of swarm intelligence. Artificial Bee Colony (ABC) Algorithm is an optimization algorithm based on the intelligent behaviour of honey bee swarm. In this work, ABC algorithm is used for optimizing different multivariable functions.

Introduction

Several modern heuristic algorithms have been developed for solving combinatorial and numeric optimization problems. While an algorithm working with a set of solutions and trying to improve them is called population based, the one using multiple iterations to approach the solution sought is named as iterative algorithm. If an algorithm employs a probabilistic rule for improving a solution then it is called probabilistic or stochastic. Another classification can be made depending on the nature of phenomenon simulated by the algorithm. This type of classification mainly has two important groups of population based algorithms: evolutionary algorithms (EA) and swarm intelligence based algorithms. The most popular EA is Genetic Algorithm (GA). GA attempts to simulate the phenomenon of natural evolution. In natural evolution, each species searches for beneficial adaptations in an everchanging environment.

Several approaches have been proposed to model the specific intelligent behaviours of honey bee swarms and applied for solving combinatorial type problems. Bee colony may be considered as a dynamical system of robots gathering information from an environment and adjusting its behaviour in accordance to it. Usually, all these robots are physically and functionally identical. The swarm possesses a significant tolerance; the failure in a single agent does not stop the performance of the whole system. The individual robots, like insects, have limited capabilities and limited knowledge of the environment. On the other hand, the swarm develops collective intelligence.

Here one approach to optimization in convex sets will be considered. In this case, completely identical robots are located inside a certain set K, and the movements are calculated as follows

$$P_{t+1} = P_t + c_1 \cdot (L - P_t) \cdot l_t + c_2 \cdot (G - P_t) \cdot g_t$$

Problem formulation

- $\mathbb{Q} \subseteq \mathbb{R}^n$ is convex
- $f : \mathbb{Q} \to \mathbb{R},\ \mathbb{Q} \subset \mathbb{R}^n$
- f is continuous on the set $\mathbb{Q}$ with the exception of the finite union of connected zero measured sets
- $x^* = \underset{x \in Q}{\mathbf{argmin}}\ f(x)$

Given to blackbox

- f - function to optimize N - the number of optiomazation steps K - the number of swarms
- $l(t),\ g(t)$ - dependencies of attraction coefficients to local and global minimum on time
- $\sigma(t)$ - dependence of random factor on time

Recommended functions is:

$$l(t) = \alpha_1 \cdot e^{-\gamma t},\ \ \mathbf{0.1 \le \alpha_1 \le 0.4}$$
$$g(t) = \mathbf{2}\alpha \cdot (\mathbf{1} - e^{-\gamma t}),\ \ \mathbf{0.05 \le \alpha_2 \le 0.3}$$
$$\sigma(t) = \beta \cdot \mathbf{diam}(\mathbb{Q}) \cdot e^{-\gamma t},\ \ \mathbf{0 \le \beta \le 0.1}$$
$$\gamma \sim N$$

Blackbox algorithm

You can see python3 code on github.

Input: f - function to minimize, $\mathbb{Q}$ - optimization area, N - number of steps, K - number of swarms, $l(t)$, $g(t)$, $\sigma(t)$ - optimize functions like a recommended

Output: $x^* \simeq \underset{x \in Q}{\mathbf{argmin}}\ f(x)$

for *i=1...K* **do**
  $bee_i \leftarrow$ random point from $\mathbb{Q}$
  $lres_i \leftarrow Bee_i\ lvalue \leftarrow f(lres_i)$
end
res $\leftarrow$ random point from Q
for *iter = 1 ... N* **do**
  for *j=1 ... K* **do**
    $c_1, c_2 \leftarrow Random([0 \dots 1])$
    $V_j \leftarrow (lres_i - bee_i) \cdot l(iter) \cdot c_1 + (res - bee_i) \cdot g(iter) \cdot c_2 + N(0, \sigma(iter))$
  end
  for *j=1 ... K* **do**
    if $bee_i + V_i \in \mathbb{Q}$ **then**
      $bee_i \leftarrow bee_i + V_i$ **if** $f(bee_i) \le f(lres_i)$ **then**
        $lres_i \leftarrow bee_i$
      end
    end
  end
  for *j=1 ... K* **do**
    $res \leftarrow min(res, lres_i)$
  end
end
return res

Algorithm 1: ABC optimizer

Results

Four functions were optimized by three global methods: Genetic Algorithm(GA), Artificial Bee Colony algorithm(ABC), Annealing Simulation(AS).
The functions are:

1. Qadratic function
$$f_1(x, y) = \|(x, y) - (0, 0)\|_2$$
2. Function with a large number of minima and semi-regular structure
$$f_2(x, y) = \mathbf{0.00005} \cdot (x^2 + y^2) + \mathbf{10}\sin(x/30) + \mathbf{10}\sin(y/30)$$
3. (Schwefels construction) Very complex function with a single global minimum
$$f_3(x, y) = \mathbf{837.9658} - x \cdot \mathbf{sin}(\sqrt{(|x|)}) - y \cdot \mathbf{sin}(\sqrt{(|y|)})$$
4. f_4 is loss function for logistic regression with polynomial (quadratic) features for two clases from popular iris dataset.

In 1-3 cases, the $\mathbb{Q} = \{x, y \in \mathbb{R} | -\mathbf{500} \le x, y \le \mathbf{500}\}$
Here is the number of iterations for custom ABC algorithm ($K = \mathbf{5}, \alpha_1 = \alpha_2 = \mathbf{0.2}$), GA from DEAP python library and AS from scipy.optimize with default paramenters. Stop criteria is $\|x - x^*\| < \mathbf{0.1}$

Method	f_1	f_1	f_3	f_4
ABC	21	85	431	1888
GA	17	42	397	913
AS	50	118	1631	7210

Table 1:Median number of iterations for different optimizers

Method	f_1	f_1	f_3	f_4
ABC	2.82	1.32	1.00	3.74
GA	3.48	3.01	6.29	1.00
AS	1.00	1.00	3.51	4.23

Table 2:Median machine time in relative units

Sample functions

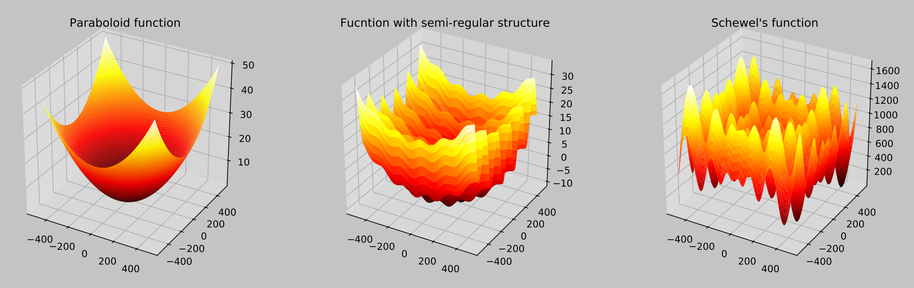

Figure 1:f_1, f_2, f_3 3d-plots

Results: Figure

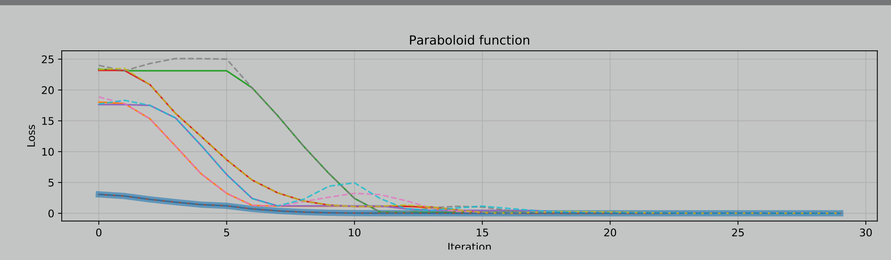

Figure 2:ABC behavior in paraboloid function

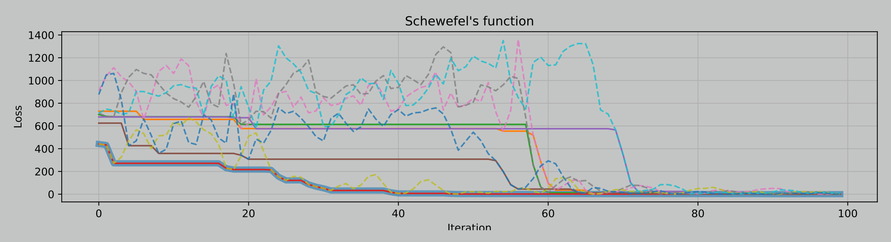

Figure 3:ABC behavior in Schewefel's function

Application

An interesting application of the algorithm is the optimization of functions defined on flows in some graphs. If in some network sources and sinks are fixed, together with their capacities, then the space of all possible flows on it will turn out to be a linear relative to the sum operator over all edges. Moreover, this remains true if certain parts of the flow are fixed, this structures are called "Semidefinite flows". There is finite-dimensional system is the space.

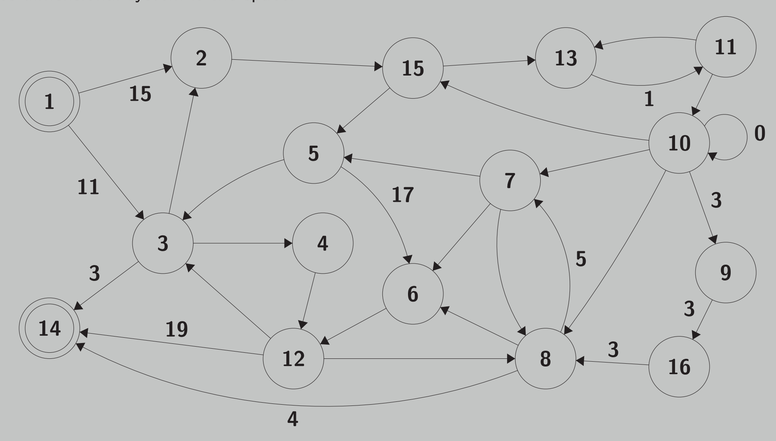

Figure 4:Example of a semidefinite flow

The problem consists in constructing a correct flow that coincides on these edges with a semidefinite flow on where some functional is minimal.
Here we will use the following loss function

$$F = \sum_{e \in \mathbf{Edges(G)}} \mathrm{Loss}_e(\mathrm{Flow}(e))$$

$$Loss_e(x) = -\mathbf{arctan}(-(x - \mu_e)) \cdot (x - \mu_e) + e^{-(x-\mu_e)}/\mu_e + e^{(x-2\mu_e)}/2\mu_e$$

Where μ_e is the edge property.

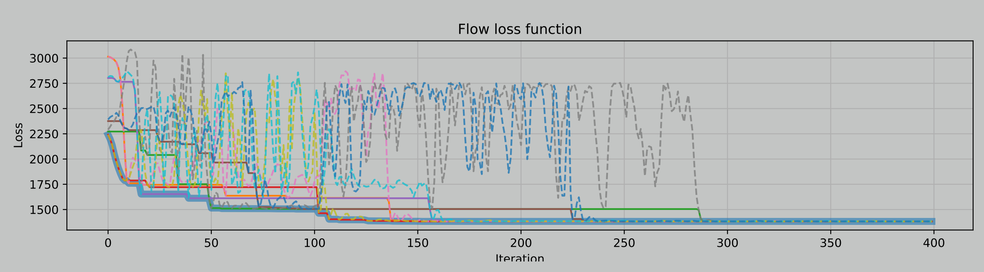

Figure 5:The typical behavior of the algorithm on a given loss function

It works :-)

Conclusion

- \+ The algorithm does not have any requirements to the smoothness of the objective function
- \+ It is possible to write a qualitative parallel implementation
- \+ It is possible to leave the local minima using appropriate parameters
- \+ Good results can be obtained by using this method to search for potential minima for further use of local optimization methods
- – The quality of work depends on the parameters that are difficult to select
- – If the parameters are incorrect, there is no chance of correct operation
- – A great waste of computer time on checking conditions if the budget set is not convex
- – In the case of convexity, the function is noticeably inferior to other methods

References

1. Pham, D.T., Karaboga, D.: Intelligent Optimisation Techniques. Springer, London (2000)
2. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)
3. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, pp. 19421948. IEEE Service Center, Piscat away (1995)
4. Vodolazsky IA, Egorov AS, Krasnov AV Roaring intellect and its most common methods of realization // Young scientist. - 2017. - 4. - P. 147-153. - URL https://moluch.ru/archive/138/38900/ (reference date: May 28, 2013).

Contact Information

- Web: https://github.com/Avi2011class
- Email: avi2011class@yandex.ru