CHEMICAL ENGINEERING TRANSACTIONS

VOL. 51, 2016

A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.aidic.it/cet

Guest Editors: Tichun Wang, Hongyang Zhang, Lei Tian

ISBN 978-88-95608-43-3; **ISSN** 2283-9216

DOI: 10.3303/CET1651082

An Improved Ant Colony Algorithm for Continuous Optimization Based on Levy Flight

Hongzhang Wang*, Conggang Liang

School of Mathematics and Information Science, Pingdingshan University, Pingdingshan, Henan 467000, China
13903759297@126.com

Ant colony optimization (ACO) is proposed on the study of the foraging behavior of ants on the basis of the proposed and widely used in the optimization. However, it has some shortcoming such as longer time, hardly implement and local optimal etc. For overcoming the above shortcoming, combined with the characteristic of Levy flight, based on Levy flight ant colony optimization is proposed which used Levy flight instead of local search for improving the searching efficiency. In order to test the performance of the new algorithm, we apply it to 20 benchmark function test and compare it with GA, PSO, ACO and LFACO algorithms. The comparison result shows that LFACO is far better than the other three algorithms in quality.

1. Introduction

With the development of computing technology, it is possible to make random search in the solution space. Random search according to the different search methods can be divided into random search and blind random search. Random search method is a feasible solution to complete random search in the solution space, and the method is very efficient when the scale of the problem is large. A guided random search rule is according to certain strategy in solution space to focus on the implementation of random search, and accumulated experience in the process of random search, makes the search is intelligent, commonly used intelligent methods such as genetic algorithm (Goldberg, 2002), particle swarm optimization (Kennedy and Eberhard, 1995).

Biologists found that by a lot of research, ant colony foraging in the process is the reason why to find the shortest path between a food source and their nest is due to ant individuals can on the way after the leave a special substance pheromones and become other ants searching for food or nest of clues. Subsequent ants encounter pheromone, not only can detect the content of pheromone, but also according to the concentration of pheromone to determine the direction of its progress. Information over time will gradually evaporate, so the length of the path and the number of residues in the number of ants on the choice of a greater impact. In a certain period of time the path by more ants, ants select the path of the greater the probability; ant chooses the path, the corresponding path information hormone concentration strengthened, to promote more ants select the path. Therefore, a large number of ants through this simple information exchange, to achieve a positive feedback of the information learning mechanism, and can quickly find the shortest path from the food source to the nest.

A typical representative of the biological colony intelligent algorithm is the ant colony algorithm. Ant Colony Optimization (ACO) is proposed on the study of the foraging behaviour of ants on the basis of the proposed (Dorigo and Caro, 1999). After years of development, the model construction algorithm relatively mature and perfect, with the characteristics of strong robustness positive feedback, distributed computing etal, has been successfully applied to discrete combinatorial optimization field, become one of the most effective means of optimization. However, ACO also has the following shortcomings:

(1) The algorithm only through the guidance of pheromone search optimization, search for a longer time;

(2) When solving the practical problem by ant colony algorithm, it is necessary to describe the problem firstly, and the algorithm is not powerful enough to describe the complex problem;

(3) In the process of searching, the algorithm is prone to search stagnation phenomenon;

Please cite this article as: Wang H.Z., Liang C.G., 2016, An improved ant colony algorithm for continuous optimization based on levy flight, Chemical Engineering Transactions, 51, 487-492 DOI:10.3303/CET1651082

In fact, it is not entirely random, but in the nature, that the individual's flight or foraging methods are not entirely random, but it obeys the Levy flight (Yang and Deb, 2009). The so-called Levy flight is a step size obeys Levy distribution random walk. Specifically, individuals generally only in a small area of flight or foraging, but there are a small fraction of an individual will suddenly fly to the distant place. This behavior is very conducive to the search process and widely used in the CS algorithm (Yang and Deb, 2010) and CSB algorithm (Yin and Liu 2015). In order to improve the search time and search stagnation, this paper proposes an ant colony algorithm based on Levy flight.
In order to Levy flight applications in more practical problems, this paper for continuous space optimization of function optimization problems, combined with the ACO algorithm, we proposed a based on Levy flight ant colony optimization algorithm (LFACO). This algorithm use Levy flight to perform local search algorithm and unlike ACO algorithm and random search, so local search ability of the algorithm has been greatly improved. In order to test the performance of the new algorithm, we apply it to 20 benchmark function test and compare it with GA, PSO, ACO and LFACO algorithms.

2. The description of LFACO

2.1 ACO algorithm

Assume that solving the problem of the scale is n, the total number of ant colony is m, the path (I, J) in time t amount of information concentration is $\tau_{ij}(t)$. An ant of K in the process of moving path selection will accord the information on each path in the pigment concentration and path of the heuristic information such as factors to calculate the state transition probability by which the ant selects the path. Here we assume that $p_{ij}^k(t)$ represents the probability that in the time t ant K is transferred from the node I to the node J, and computing the probability calculation is:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha * [\mu_{ij}(t)]^\beta}{\sum_{s \subset allowed_k} [\tau_{is}(t)]^\alpha * [\mu_{is}(t)]^\beta}, if\ j \in allowed_k \\ 0, otherwise \end{cases} \quad (1)$$

In the above equation, $allowed_k$ represents a collection of the ant K step allowing the selection of node. α is the pheromone heuristic factor, β is expected heuristic factor, $\mu_{ij}(t)$ is the heuristic function. In order to avoid earlier information residues too much and cause residue information flooded the heuristic information of the phenomenon, in each ant finish or complete traversal of all n cities, to update operation treatment of residual information. So in the t+1 time the pheromone concentration of the path (I, J) can be adjusted and updated as the following rules:

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t) \quad (3)$$

In the above equations, ρ is the pheromone evaporation coefficient, and for preventing unlimited information accumulation, the range of P is [0, 1). $\Delta\tau_{ij}(t)$ is said as the path (I, J) pheromone increment, $\Delta\tau_{ij}^k(t)$ is said the information concentration the kth ant after path (I, J). So the pseudo code of ACO is as Figure 1.

```
The Ant Colony Optimization
Step 1: Set parameters, initialize pheromone trails
Step 2: While termination condition not met do
        Step 2.1 Construct Ant Solutions
        Step 2.2 Apply Local Search (optional)
        Step 2.3 Update Pheromones
End while
```

Figure 1: The pseudo code of ACO

2.2 Levy Flight

Animals foraging path was considering as a random or quasi-random manner in nature. However, various studies have shown that the flight behavior of many animals and insects obeys the typical characteristics of Levy flights. Broadly speaking, Levy flights are a random walk whose step length is drawn from the Levy distribution, often in terms of a simple power-law formula L(s)~|s|^(-1-β) where 0<β<2. Obviously, the generation of step sizes samples is not trivial using Levy flights. A simple scheme discussed in detail can be summarized as following (Yang, 2010):

$$L(s) \sim \frac{u}{|v|^{\frac{1}{\beta}}} \tag{4}$$

In which $u \sim N(0, \sigma_u^2)$, $v \sim N(0, \delta_v)$ are normal distribution.

2.3 The description of LFACO

For improving the performance of ACO, a possible method is to change the pheromone volatilization coefficient ρ using Levy distribution while ρ is a constant in ACO, so we propose an new improved ACO algorithm based on Levy Flight (LFACO) in which each movement of each ant individual obey the levy distribution. In LFACO, the equation (4) is introduced into the equation (2) and gets the following equation for updating the location of each ant individual:

$$\tau_{ij}(t+1) = \left(1 - \rho_{i,j}\right) * \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{5}$$

In the eq(5), $\rho_{i,j} \sim levy(\beta)$, $levy(\beta) \sim \frac{u}{|v|^{\frac{1}{\beta}}}(\tau_{ij}(t+1) - \tau_{ij}(t))$ *and* $u \sim N(0, \sigma_u^2)$, $v \sim N(0,1)$, $\sigma_u = \frac{\mathcal{T}(1+\beta)sin(\beta\pi/2)}{\mathcal{T}\left(\frac{1+\beta}{2}\right)\beta 2^{(\beta-1)/2}}^{1/\beta}$.Combined the above analysis, the step of LFACO is following as figure 2.

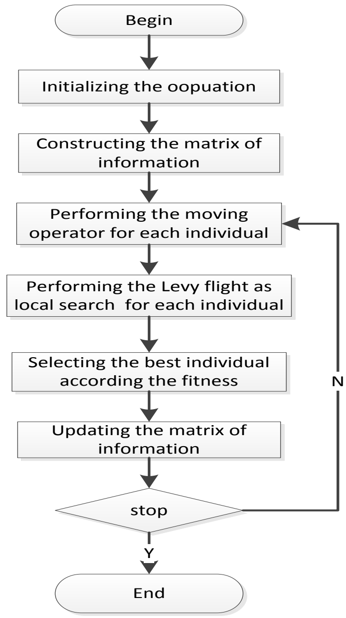

Figure 2: The description of LFACO.

Function Optimization

Function Optimization is often expressed in the following form:

$$min\, f(X) \quad subject\ to \quad L \leq X \leq U \tag{6}$$

In which $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \dots, \mathrm{x}_\mathrm{n}\}$ is a vector, and $\mathrm{L} = \{\mathrm{l}_1, \mathrm{l}_2, \mathrm{l}_3, \dots, \mathrm{l}_\mathrm{n}\}$ is the lower bound of X, while $\mathrm{U} = \{\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3, \dots, \mathrm{u}_\mathrm{n}\}$ is the upper bound of X. In this work, we test and verify the effectiveness of the proposed LFQPSO algorithm in this chapter by using the following 15 benchmark functions of which its definition, domains and optimal value are respectively defined as:

(1)The definition of f_1 is following:

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{7}$$

Its domains is $-5.12 \leq \mathrm{x_i} \leq 5.12, \mathrm{i} = 1,2,3, \dots, \mathrm{n}$

Its argument and optimal value are $\mathrm{x}^* = (0,0,0, \dots, 0), \mathrm{f}(\mathrm{x}^*) = 0$

(2)The definition of f_2 is following:

$$f_2(x) = \sum_{i=1}^{n}(ix_i^2) \tag{8}$$

Its domains is $-5.12 \leq x_i \leq 5.12, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(3)The definition of f_3 is following:

$$f_3(x) = \sum_{i=1}^{n-1}[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \quad (9)$$

Its domains is $-2.048 \leq x_i \leq 2.048, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (1,1,1,\dots,1), f(x^*) = 0$
(4)The definition of f_4 is following:

$$f_4(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i)) \quad (10)$$

Its domains is $-5.12 \leq x_i \leq 5.12, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(5)The definition of f_5 is following:

$$f_5(x) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (11)$$

Its domains is $-1 \leq x_i \leq 1, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(6)The definition of f_6 is following:

$$f_6(x) = \sum_{i=1}^{n}|x_i|^{i+1} \quad (12)$$

Its domains is $-32.768 \leq x_i \leq 32.768, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(7)The definition of f_7 is following:

$$f_7(x) = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}cos(2\pi x_i)} \quad (13)$$

Its domains is $-32.768 \leq x_i \leq 32.768, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(8)The definition of f_8 is following:

$$f_8(x) = max(|x_i|), i = 1,2,3,\dots,n \quad (14)$$

Its domains is $-100 \leq x_i \leq 100, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(9)The definition of f_9 is following:

$$f_9(x) = max(|x_i|), i = 1,2,3,\dots,n \quad (15)$$

Its domains is $-100 \leq x_i \leq 100, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(10)The definition of f_{10} is following:

$$f_{10}(x) = \sum_{i=1}^{n}x_i^2 + (\sum_{i=1}^{n}0.5ix_i)^2 + (\sum_{i=1}^{n}0.5ix_i)^4 \quad (16)$$

Its domains is $-5 \leq x_i \leq 10, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(11)The definition of f_{11} is following:

$$f_{11}(x) = sin^2(\pi y_1) + \sum_{i=1}^{n-1}[((y_i - 1)^2(1 + 10sin^2(\pi y_i + 1)))] + (y_n - 1)^2(1 + sin^2(2\pi y_n)), y_i = 1 + \frac{x_i - 1}{4}, i = 1,2,3,\dots,n \quad (17)$$

Its domains is $-10 \leq x_i \leq 10, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (1,1,1,\dots,1), f(x^*) = 0$
(12)The definition of f_{12} is following:

$$f_{12}(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i}x_j^2) \quad (18)$$

Its domains is $-65.536 \leq x_i \leq 65.536, i = 1,2,3,\dots,n$
Its argument and optimal value are $x^* = (0,0,0,\dots,0), f(x^*) = 0$
(13)The definition of f_{13} is following:

$$f_{13}(x) = 1 - cos\left(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2} \quad (19)$$

Its domains is $-100 \leq x_i \leq 100, i = 1,2,3, \dots, n$
Its argument and optimal value are $x^* = (0,0,0, \dots ,0), f(x^*) = 0$
(14)The definition of f_{14} is following:

$$f_{14}(x) = 0.1 + \sum_{i=1}^{n}(sin(x_i))^2 - 0.1\prod_{i=1}^{n}(exp(-x_i^2)) \quad (20)$$

Its domains is $-10 \leq x_i \leq 10, i = 1,2,3, \dots, n$
Its argument and optimal value are $x^* = (0,0,0, \dots ,0), f(x^*) = 0$
(15)The definition of f_{15} is following:

$$f_{15}(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1} + \frac{n(n+4)(n-1)}{6} \quad (21)$$

Its domains is $-n^2 \leq x_i \leq n^2, i = 1,2,3, \dots, n$
Its argument and optimal value are $x^* = i(n - i + 1), i = 12,3, \dots, n, f(x^*) = 0$

4. Experiment Result

In order to compare the results of LFACO algorithm, GA, PSO, ACO and LFACO algorithm use the R programming language implementation, and all the program results in 3.3GHz Core Duo processor, 4GB of ram and windows 7 operating system of PC operation. For the 15 benchmark functions, the four algorithms are run 100 times respectively, and the average experimental results are shown in Table 1 and table 2.
As Table 1 shown, for all 20 benchmark test functions, the convergence speed of PSO is the best, and of LFACO is better than that of GA and ACO, which shows that LFACO can indeed faster than ACO in optimizing the optimal searching process.
Next we focus on analysis the result of LFACO and the rest of the three algorithms. Table 2 shown the result is the best LFACO optimization, and although the execution time of the PSO algorithm best as Table 1 shown, but the analysis table 2, we can see that PSO result is the worst in four algorithms, which shows the no an algorithm in which some indexes can reflect the best.

Table 1: Runtime of GA, PSO, ACO and LFACO on the 15 benchmark functions

Function	GA	PSO	ACO	LFACO
f_1	10.41	2.21	10.21	7.34
f_2	10.74	2.28	10.53	7.57
f_3	11.04	22.64	10.82	7.78
f_4	11.67	2.48	11.45	8.23
f_5	11.49	23.58	11.27	8.10
f_6	10.39	2.44	10.73	8.10
f_7	11.84	2.51	11.62	8.35
f_8	10.71	2.27	10.50	7.55
f_9	10.84	2.30	10.63	7.64
f_{10}	11.18	2.37	10.97	7.89
f_{11}	11.50	2.44	11.28	8.11
f_{12}	10.82	2.30	10.61	7.63
f_{13}	10.96	2.33	10.75	7.73
f_{14}	11.49	2.44	11.27	8.10
f_{15}	11.42	2.42	11.20	8.05

5. Conclusion

In order to improve performance ACO algorithm in function optimization, in this paper the Levy flight introduced to ACO algorithm and get the new update of the individual equations. Based on the above discuss, we propose the new algorithm of based on Levy Flight of the ACO algorithm (LFACO) and detailed describe its steps and processes. In order to verify the LFACO algorithm, 15 benchmark functions are used to test the

LFACO which is compared with GA, PSO algorithm and ACO algorithm. The comparison results show that for the function optimization problem LFACO in effectiveness as well as run time is far better than other three algorithms.

Table 2: the average and standard variation of GA, PSO, ACO and LFACO on the 15 benchmark functions

Function	GA	PSO	ACO	LFACO
f_1	0.00±0.06	0.00±8.06	2.05±1.29	0.00±0.00
f_2	0.16±0.37	5.43±16.97	87.13±20.54	0.00±0.00
f_3	28.71±0.96	44.12±34.10	84.24±32.06	12.67±2.97
f_4	15.80±4.69	97.41±32.9	99.27±13.52	3.01±1.48
f_5	0.67±0.19	0.03±2.67	208.24±39.01	0.00±0.00
f_6	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
f_7	1.86±0.53	0.01±3.41	8.25±0.84	0.04±0.03
f_8	12.43±2.46	17.91±5.35	24.08±2.52	4.90±1.11
f_9	0.25±0.30	80.00±23.41	46.99±8.60	0.00±0.00
f_{10}	27.32±4.18	86.19±14.73	94.91±101.01	0.68±0.60
f_{11}	0.01±0.35	1.85±5.09	4.17±13.83	0.00±0.00
f_{12}	11.32±77.17	258.88±28.96	139.01±31.99	0.00±0.00
f_{13}	3.30±0.50	0.82±2.06	6.00±0.97	1.80±0.32
f_{14}	0.13±0.05	3.36±0.75	0.23±0.08	0.10±0.00
f_{15}	222.33±163.57	192.27±125.16	122.36±43.15	82.83±33.85

Acknowledgment

Fund Project: 2011 Henan provincial science and Technology Department of science and technology foundation project; Exact solutions of higher order nonlinear partial differential equation; Project approval number: 112300410199.

Reference

Dorigo M., Caro G.D., 1999, Ant colony optimization: A new meta-heuristic, Proceedings of the 1999 Congress on Evolutionary Computation, 2:1470-1477.

Goldberg D.E., 2002, the Design of Innovation: Lessons from and for Competent Genetic Algorithms. Norwell, MA: Kluwer Academic Publishers. ISBN 978-1402070983

Kennedy J., Eberhart R., 1995, Particle swarm optimization, IEEE International Conference on Neural Networks, 4, 1942-1948, DOI: 10.1109/ICNN.1995.488968.

Yang X.S., Deb S., 2009, Cuckoo Search via lévy flights. Proceeding of world congress on Nature & Biologically Inspired Computing, India, 210-214.

Yang X.S., Deb S., 2010, Engineering optimization by Cukcoo Search, Int. J. Mathematical Modeling and Numerical Optimization, 1(4), 330-343.

Yin L., Liu Y.G., 2015, Biclustering of the gene expression data by coevolution cuckoo search. International Journal Bioautomation, 19(2): 161-176.