A gravitational search algorithm based on levy flight

Jing Zhao a, Haidong Zhu b, Yinhua Hu c, Enjun Hu d, Baole Huang e, Tingyu Zhang f and Pan Zhang g

Nanjing Huadun Electric power Information Security Assessment Co., Ltd, China

a Zhaojing_seu@hotmail.com, b 171171531@qq.com, c yinhua-hu@sac-china.com, d hej-enjun.hu@sac-china.com, e 996252145@qq.com, f 960528095@qq.com, g zpanjing@163.com

Abstract. A gravitational search algorithm based on levy flight is proposed to solve the problem that the gravitational search algorithm falls into the local optimal solution. Firstly, the dynamic adjustment strategy of gravitational constant is introduced to increase the diversity of particles and improve the convergence speed and accuracy of the algorithm. Then, on the basis of gravitation search algorithm, levy flight is introduced, and the particle search range is increased by using levy flight search mode combining small step and long stride search, so as to improve the global search capability of the algorithm. Finally, six standard test functions are used for simulation analysis. The results show that the improved algorithm has obvious advantages in both convergence speed and convergence precision, and it is more advantageous in multi-peak function optimization.

Keywords: Gravitational search algorithm; levy flight; gravitational constant; function test.

1. Introduction

The gravitational search algorithm [1] (GSA) was proposed in 2009 by Esmat Rashedi, a professor at Iran’s Kerman University. The algorithm is a meta-heuristic intelligent optimization algorithm based on Newton’s law of gravitation. Compared with other intelligent optimization algorithms, the GSA has the advantages of simple structure, fewer parameters, and strong global optimization capabilities. Particle swarm optimization algorithm [2], genetic algorithm [3], ant colony algorithm [4] and difference compared with the evolutionary algorithm [5], the optimization ability and algorithm accuracy of the GSA algorithm have obvious advantages. At present, experts and scholars have verified the ability of this algorithm in optimization through experimental research [6].

Although GSA has certain advantages compared to other intelligent calculation methods, the basic gravitational search algorithm still has problems such as premature convergence, poor local search ability, and lack of effective acceleration mechanisms [7]. In view of the shortcomings of GSA, many scholars have proposed improvements. Literature [8] uses chaos strategy for local search to improve the pioneering nature of the algorithm and improve the optimization accuracy of the algorithm. Literature [9] made a forbidden conversion to the universal gravitation algorithm, and proposed a binary gravitation search algorithm. Literature [10] puts forward the combination of binary particle swarm...
algorithm and binary gravity algorithm based on the combination of particle swarm algorithm and gravity algorithm, which makes the mitigation algorithm better in optimization. Literature [11] proposed a gravitational search algorithm for adaptive chaotic mutation. The average particle distance was introduced into the algorithm to control the degree of dispersion of particle mutual distribution, and chaotic mutation [12] was used to mutate the position of particles. The boundary variation processing method is adopted for the particles that cross the boundary, which has better convergence performance. Literature [13] introduced Logistic mapping into GSA, replaced random sequences with chaotic sequences generated by logistic mapping, and then used chaos as a local search method.

The above improved algorithms have enhanced the optimization capabilities of the GSA from different angles, and the performance has been greatly improved, and has been widely used, but how to better balance the global exploration and local development capabilities still needs further research. In view of this, this article proposes a gravitational search algorithm based on levy flight (LGSA). First, introduce the dynamic adjustment strategy of the gravitational constant G to increase the diversity of particles and improve the convergence speed and accuracy of the algorithm; then, the levy flight is introduced on the basis of the universal gravitation search algorithm, and the levy flight is small and large. The search mode combined with search increases the search range of the particles, improves the algorithm's global search capability, and further improves the algorithm's convergence speed and accuracy; Finally, simulation analysis is performed through 6 standard test functions to verify the effectiveness of the LGSA.

2. Gravitational search algorithm
The universal gravitation search algorithm is an intelligent optimization method based on the law of universal gravitation. The principle of the algorithm is to treat the search particles as a group of objects running in space, and the movement of the objects follows the law of dynamics. The larger the moderate value of the particle, the greater the inertial mass, and the universal gravitation will prompt the objects to move toward the object with the largest mass, thereby gradually approaching the optimal solution of the optimization problem [14]. According to the review, the GSA process is described as follows:

Step1: Initialize the relevant parameters of the algorithm. Assume randomly generate N particles in the D dimensional search space, where the position of the ith particle can be expressed as:

$$X_i = (x_i^1, x_i^2, ..., x_i^d), \ i = 1, 2, ..., N$$  \hspace{1cm} (1)

Where $x_i^k$ denotes the kth dimension of the ith particle.

Step2: Update particle position and velocity.

$$a_i^d(t) = F_i^d(t) / M_i(t)$$

$$v_i^d(t + 1) = rand_i \times v_i^d(t) + a_i(t)$$

$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1)$$  \hspace{1cm} (2)

Where $a_i^d(t), x_i^d(t), v_i^d(t)$ represent the acceleration, position and velocity of the Dth dimension of the ith particle; $rand_i$ is a random number between [0,1], which makes the search more reasonable with a certain randomness.

Step3: Calculate the resultant force of particles.

$$F_i^k(t) = \sum_{j=1, j \neq i}^{N} rand_j F_{ij}^k(t)$$  \hspace{1cm} (3)

$$F_{ij}^k(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \epsilon} (x_i^k(t) - x_j^k(t))$$  \hspace{1cm} (4)

Where N is the total number of examples; $F_{ij}^k(t)$ is the gravitation between particles of jth to ith; $rand_j$ is a random number between [0,1]; $R_{ij}(t)$ is the euclidean distance between particles of jth to ith.
\[ R_0(t) = \| X_i(t), X_j(t) \| \]  

Step 4: \( G(t) \) represents the gravitational constant at time \( t \):
\[ G(t) = G_0 \times e^{-\alpha t/T} \]  

Where \( G_0 = 100; \alpha = 20; T \) is the maximum number of iterations. 

Step 5: Calculation of particle inertia mass \( M_i(t) \).
\[
\begin{cases}
    m_i(t) = \frac{(f_i(t) - \text{worst}(t))}{(\text{best}(t) - \text{worst}(t))} \\
    M_i(t) = m_i(t) / \sum_{j=1}^{N} m_i(t)
\end{cases}
\]  

Where \( f_i(t) \) represents the fitness value of the \( i \)th particle at time \( t \). 

Step 6: Calculate the worst and best solution of particle.
\[
\begin{cases}
    \text{best}(t) = \max_{i \in \{1,2,...,N\}} f_i(t) \\
    \text{best}(t) = \min_{i \in \{1,2,...,N\}} f_i(t)
\end{cases}
\]  

3. Gravitational search algorithm based on levy flight

3.1. Dynamic adjustment of gravitational constant

The gravitational constant directly affects the resultant force and acceleration of the particles, and it has an important impact on whether the particles can jump out of the local optimum in GSA and improve the accuracy of the optimal solution. A reasonable setting of the gravitational constant can ensure the search performance of GSA.

It can be seen from (6), that the magnitude of the gravitational constant decreases rapidly with the increase of \( t \) at the beginning of the iteration, when \( t \) is large enough, it tends to be stable, and it decreases faster in the early stage of the iteration and slower in the end of the iteration. This paper introduces the gravitational constant dynamic adjustment strategy to design the gravitational constant.

\[ G'(t) = G_0 \times e^{-\alpha_1 t^\alpha_2} \]  

Where \( \alpha_1 = 1; \alpha_2 = 1.5 \), which is mainly based on experience.

Comparing with Fig. 1, it can be seen that \( G'(t) \) takes large steps at the beginning of the iteration, which is beneficial to control the step length to prevent particles from jumping over points with high fitness values, increase the diversity of particles while maintaining good search capabilities. The end of the iteration is conducive to the refinement and optimization of the particles, enhances the development performance of the particles, effectively avoids the premature convergence of the particles and falls into the local optimum, and improves the optimization accuracy.
3.2. Levy Flight

Levy flight is a non-Gaussian random process, a random search mode that obeys Levy distribution[16]. It is a search mode that combines small step search and occasional large step search. It can prove many random phenomena in nature, such as random walking and brownian motion. Because Levy flight has the characteristics of power rate distribution and generalized central limit theorem, the levy flight mode is adopted in the algorithm, which increases the search range of particles and is not easy to fall into the local optimum, which is beneficial to solving optimization and optimization problems.

The update formula of levy flight position is:

\[
x_{i}^{(t+1)} = x_{i}^{(t)} + \alpha \odot \text{Levy}(\lambda) \quad i = 1, 2, \ldots, n
\]

(10)

Where \(\oplus\) is point-to-point multiplication; \(\text{Levy}(\lambda)\) is Random search path:

\[
\text{Levy}(\lambda) = 0.01 \times \frac{u \cdot \phi}{|v|^\beta}
\]

(11)

Where \(u, v\) obey normal distribution; \(\beta\) is a variable that controls the distribution, generally between \([0,2]\); \(\phi\) is variance:

\[
\phi = \begin{cases} 
\Gamma(1 + \beta) \times \sin \left( \frac{\pi \times \beta}{2} \right) \times 2^{-\frac{1}{\beta}} \\
\Gamma \left( \frac{1 + \beta}{2} \right) \times \beta \times 2^{-\frac{\beta - 1}{2}} \end{cases}
\]

(12)

In order to reflect the superiority of levy flight, by using MATLAB to simulate the random walk and levy flight trajectory, both steps are set to 200 steps, as shown in Fig. 2. It can be seen from the figure that levy flight has a broader search range and ability. Applying this feature to the gravitational search algorithm can effectively improve the vitality and jumping ability of the search particles, thereby improving the algorithm’s global optimization ability.

![Fig. 2 Levy flight and random walk](image)

4. Experimental results and analysis

4.1. Test environment and parameter settings

The test environment of this paper is as follows: the CPU of the computer is Intel(R) Core(TM) i5-3230M CPU@2.60GHZ, the memory is 6.0GB; the software environment is Windows 7 operating system and MATLAB 2015b version.

In this experiment, we set the population size \(N = 20\), the maximum number of iterations \(T = 100\), the gravitational constant \(G_0 = 100\), and the decay rate \(\alpha = 20\).
4.2. Test function
In order to verify the feasibility and superiority of the proposed LGSA proposed in this paper, six different types of benchmark functions were selected for simulation experiments, as shown in Table 1. Function f1 is a nonlinear, smooth, symmetric unimodal function, f2 is a typical ill-conditioned banana-type function; f3–f6 is a typical nonlinear Multi-peak function, these four functions all introduce trigonometric functions, which make the function surface have a large number of peaks and valleys.

| Function name | Function expression | Search space | Global optimum |
|---------------|---------------------|--------------|----------------|
| Sphere        | $f_1(x) = \sum_{i=1}^{D} x_i^2$ | $[-5.12,5.12]$ | 0.00 |
| Schewefel     | $f_2(x) = \sum |x_i| + \prod |x_i|$ | $[-10,10]$ | 0.00 |
| Griewank      | $f_3(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$ | $[-600,600]$ | 0.00 |
| Schaffer      | $f_4(x) = 0.5 + \frac{\sin \left( \sqrt{x_i^2 + x_j^2} \right)}{1 + 0.001(x_i^2 + x_j^2)} - 0.5$ | $[-10,10]$ | 0.00 |
| Ackley        | $f_5(x) = -c_i \exp \left\{ -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right\} - \exp \left\{ -0.2 \frac{1}{D} \sum_{i=1}^{D} \cos \left( 2\pi x_i \right) \right\} + c_i + e$ | $[-32,32]$ | 0.00 |
| Rastrigin     | $f_6(x) = \sum_{i=1}^{D} \left[ x_i^2 - 10 \cos \left( 2\pi x_i \right) + 10 \right]$ | $[-5.12,5.12]$ | 0.00 |

4.3. Analysis of results
In order to measure the superiority of the LGSA compared with the GSA algorithm, this paper selects the average value, standard deviation, optimal solution and worst solution of the optimization results as the evaluation index. In fact, considering that intelligent optimization algorithms generally have a certain degree of randomness, the experiments are run independently 100 times, the calculation results are shown in Table 2, and the distribution of the results is shown in Fig. 3.

| Algorithm | function index | Sphere | Schewefel | Griewank | Schaffer | Ackley | Rastrigin |
|-----------|----------------|--------|-----------|----------|----------|--------|-----------|
| LGSA      | Mean 5.80E-36  | 1.59E-16 | 0.00      | 0.00     | 9.24E-16 | 0.00   |
|           | Std 5.18E-35   | 1.58E-15 | 0.00      | 0.00     | 3.54E-16 | 0.00   |
|           | Worst 5.20E-34 | 1.58E-14 | 0.00      | 0.00     | 4.44E-15 | 0.00   |
|           | Best 4.79E-50  | 5.91E-26 | 0.00      | 0.00     | 8.88E-16 | 0.00   |
|           | Time 6.83      | 7.38    | 7.98      | 7.50     | 7.70     | 7.99   |
| GSA       | Mean 2.84E-18  | 3.09E-02 | 5.94      | 8.53E-03 | 6.92     | 8.16E-01|
|           | Std 3.07E-18   | 0.1680  | 6.1521    | 5.53E-03 | 5.4356   | 8.94E-01|
|           | Worst 1.37E-17 | 1.45    | 3.80E+01  | 3.72E-02 | 1.80E+01 | 4.97   |
|           | Best 5.71E-21  | 2.98E-10 | 1.26E-01  | 0.00     | 2.40E-08 | 0.00   |
|           | Time 6.01      | 6.19    | 6.29      | 6.47     | 6.11     | 6.19   |
Observing Fig. 3 and Table 2, we can find that the LGSA is more effective than the GSA for the same test function. The average value, standard deviation, optimal solution, and worst solution obtained by optimization are significantly better than GSA. For the unimodal function Sphere function and the Schewefel function, the optimization accuracy of the LGSA is 18 orders of magnitude and 14 orders of magnitude higher than that of the GSA, respectively; for multimodal functions with fewer local optimal values, such as Ackley function, compared with the GSA, the optimization accuracy is improved by 16 orders of magnitude. For multimodal functions with more local optimal values, such as Griewank function, Schaffer function and Rastrigin function, the LGSA has obvious optimization effects and directly converges to the optimal solution. Based on the above analysis, it can be known that for different test functions, the LGSA is significantly better than the GSA in the optimization effect and stability, and the optimization performance of the LGSA is more obvious for the optimization effect of the complex test function. The GSA is easy to fall into the local optimal solution in the optimization process, and the LGSA uses the search mode of the levy flying small step size and the large step size search to help the particles easily jump out of the local optimal solution, thereby finding the global optimal solution value.

In addition, it can be seen from Table 2 that the stability, reliability, and optimization accuracy of the LGSA are significantly higher than the GSA, but its overall running time is higher than the GSA, mainly due to the fact that the added levy flight has the intermittent and irregular search characteristics result in a slower update speed of individuals and an increase in running time. In order to further illustrate that the LGSA also has great advantages in convergence speed, the convergence curve of each algorithm
when the number of particles \( N = 50 \), the dimension \( D = 30 \), and the number of iterations is 100 is given. The test results are compared as shown in Fig. 4.

![Graphs showing test function convergence curves for different functions](image)

Fig. 4 Test function convergence curve

It can be seen from Fig. 4 that for the six test functions, LGSA requires the least number of iterations and the highest convergence accuracy compared to the GSA to converge to a certain accuracy. In the early iterations, the initial value of the LGSA is significantly better than the GSA, mainly due to the dynamic adjustment of the gravitational constant \( G \), which enables the particles to skip points with higher fitness values in the early iterations, increasing the diversity of particles while maintaining good search ability. For the Sphere function and Schewefel function, LGSA and GSA can converge to the optimal solution better, but the convergence speed of LGSA is the fastest, indicating that for unimodal functions like Sphere function and Schewefel function, the convergence performance of LGSA is obviously better than that of the GSA; For multimodal functions, the convergence speed and accuracy of the LGSA are significantly better than the GSA. The GSA is very easy to fall into a local optimal solution, mainly due to the introduction of levy flight by LGSA, which is easier for multimodal functions to jump out of the local optimal solution.
5. Conclusion
Based on the analysis of the optimization principle of the GSA algorithm and the mathematical properties of levy flight, this paper introduces the dynamic adjustment strategy of the gravitational constant G to increase the diversity of particles and improve the convergence speed and accuracy of the algorithm. At the same time, the levy flight with jumping characteristics is introduced. Added to the GSA, the algorithm has stronger vitality during the position update process, can effectively expand the search range, and make it easier for the particles to jump out of the local optimal position, thereby better improving the convergence performance and optimization of the algorithm Accuracy.

References
[1] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A Gravitational Search Algorithm[M]. 2009.
[2] Yu K, Wang X, Wang Z. Multiple learning particle swarm optimization with space transformation perturbation and its application in ethylene cracking furnace optimization[J]. Knowledge-Based Systems, 2016, 96:156-170.
[3] Chuang Y C, Chen C T, Hwang C. A real-coded genetic algorithm with a direction-based crossover operator[J]. Information Sciences, 2015, 305:320-348.
[4] Yang Q, Chen W N, Yu Z , et al. Adaptive Multimodal Continuous Ant Colony Optimization[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(2):191-205.
[5] Cheng J , Zhang G , Caraffini F , et al. Multicriteria adaptive differential evolution for global numerical optimization. [J]. Integrated Computer-Aided Engineering, 2015, 22(2):103-107.
[6] Li C , Zhou J . Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm[J]. Energy Conversion and Management, 2011, 52(1):374-381.
[7] Peng Li, Weina Xu, Zeyuan Zhou, et al. Optimal Operation of Microgrid Based on Improved Gravitational Search Algorithm [J]. Proceedings of The Chinese Society for Electrical Engineering, 2014, 34(19):3073-3079.
[8] Gao S , Vairappan C , Wang Y , et al. Gravitational search algorithm combined with chaos for unconstrained numerical optimization[J]. Applied Mathematics and Computation, 2014, 231:48-62.
[9] Rashedi E, Nezamabadi-Pour H, Saryazdi S. BGSA: binary gravitational search algorithm[J]. Natural Computing, 2010, 9(3):727-745.
[10] Mirjalili S , Wang G G , Coelho L D S . Binary optimization using hybrid particle swarm optimization and gravitational search algorithm[J]. Neural Computing and Applications, 2014, 25(6):1423-1435.
[11] Ping Luo, Wei Liu, Shubo Zhou. Gravitational Search Algorithm of Adaptive Chaos Mutation [J]. Journal of Guangdong University of Technology, 2016(1):57-61.
[12] Yong Liu, Liang Ma. Solving nonlinear minimax problems based on chaos gravitational search algorithm [J]. Application Research of Computers, 2012, 29(1).
[13] Gao S , Vairappan C , Wang Y , et al. Gravitational search algorithm combined with chaos for unconstrained numerical optimization[J]. Applied Mathematics and Computation, 2014, 231:48-62.
[14] Zibanezhad B , Zamanifar K , Sadjady R S , et al. Applying gravitational search algorithm in the QoS-based Web service selection problem[J]. Frontiers of Information Technology & Electronic Engineering, 2011, 12(9):730-742.
[15] Xie J, Zhou Y, Chen H. A novel bat algorithm based on differential operator and levy flights trajectory[J]. Computational Intelligence and Neuroscience,2013,(2013-3-17), 2013, 2013:453812.
[16] Rubenthaler S. Numerical simulation of the solution of a stochastic differential equation driven by a Lévy process[J]. Stochastic Processes & Their Applications, 2003, 103(2):311-349.