Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 4, April 2018, pp. 681-690
DOI: 10.23940/ijpe.18.04.p11.681690

A Gravitational Search Algorithm with Adaptive Mixed Mutation for Function Optimization

Jingsen Liu[a,b], Yuhao Xing[b], and Yu Li[c,*]

[a]*Institute of Intelligent Network system, Henan University, Kaifeng, 475004, China*
[b]*College of Software, Henan University, Kaifeng, 475004, China*
[c]*Institute of Management Science and Engineering, Henan University, Kaifeng, 475004, China*

Abstract

In order to improve the optimization accuracy and convergence speed of gravitational search algorithm, the gravitational search algorithm with the mechanism of adaptive mixed random mutation is proposed. A mutation trigger function with adaptive property is introduced into the algorithm, so that every particle has the probability of mutation at any time, and the number of particles that change in the population tends to decrease with the increase of iteration times. At the same time, in the whole optimization process of the algorithm, the uniform mutation and Laplace-normal hybrid mutation cooperate together. The uniform mutation enables the algorithm to find the global optimal area quickly, and then continue deep search with hybrid mutation to improve the optimization performance. The simulation results show that in solving the problem of extremum optimization, the improved algorithm has significantly improved optimization performance, and has high convergence accuracy and faster convergence speed.

Keywords: gravitational search algorithm; trigger function of mutation; mixed mutation; function optimization; optimization precision; convergence

(Submitted on January 10, 2018; Revised on February 24, 2018; Accepted on March 27, 2018)

1. Introduction

With the development of science and technology, combinatorial optimization is becoming more and more complex. The traditional numerical method is not quick and convenient in solving large-scale optimization problems. In order to solve the problem, researchers began to seek a new algorithm. They put forward a series of heuristic algorithms such as ant colony algorithm for simulating ant collective behavior, particle swarm optimization for simulating the behavior of bird population, genetic algorithm for simulating the mechanism of biological evolution in nature, tabu search algorithm for simulating the memory process of human intelligence, simulated annealing algorithm for simulated annealing process of solid matter, bat algorithm to simulate bats echolocation and so on. These heuristic algorithms can get better results than traditional numerical methods in solving various complex optimization problems.

Gravitational search algorithm [14] (GSA) is a new heuristic global optimization algorithm proposed by Professor E. Rasheeda. The algorithm simulates the most common gravitational force phenomena in nature. Individuals are attracted by the action of the gravitational force and move towards the direction of the largest mass of individuals. The motion of individuals follows Newton's second law. With the continuous movement, eventually the whole population will gather around the largest quality individuals. In this way, the largest quality individuals can be obtained and the best individuals occupy the best position; the optimal solution of the problem is achieved. At present, the algorithm has been successfully used to solve problems about choosing Web server of QoS [18] based on filtering model of linear and nonlinear classification [13], classification data [2], estimates the decision function [1], fault diagnosis [15], identification of parameters of the chaotic system [6], multi objective economic decision [12], slope stability analysis [4] etc.

* Corresponding author.
E-mail address: lyhenu@163.com

On one hand, gravitational search algorithm has the advantages of swarm intelligence algorithm such as strong global search ability and fast convergence speed. But there are some shortcomings such as premature maturity and low accuracy. Aiming at these shortcomings, many scholars have made some improvements on gravitational search algorithm. For example, in reference [5], fuzzy rules are adopted to make nonlinear dynamic adjustment of the gravitational coefficient, which effectively avoids the problem that standard gravitational search algorithm is easy to fall into local optimum. In reference [16], by constructing elite search strategy and diversity enhancement mechanism, the speed of each particle moving to a new position is controlled and the algorithm is effectively prevented from precocity. In reference [3], the global optimal solution of particle swarm optimization and the idea of local optimal solution are applied to the gravitational search algorithm. The convergence of the algorithm is improved. Reference [11] makes use of the best position and adds dynamic learning factor to make the particle move faster to the global optimal region, and improve the optimization performance of the algorithm. These improvements have improved the performance of the algorithms in their respective applications, but the gravitational search algorithm still has further improvement. This paper proposes gravitational search algorithm with a novel the mechanism of adaptive mixed random mutation (MGSA), used to solve the extreme value of function optimization problem. According to iterative evolution, the algorithm adjusts the mutation of the number of individuals dynamically in the population and uses the uniform mutation and Laplace-normal hybrid mutation, so that the population optimization precision has been improved obviously. The experimental results show that the improved algorithm has higher optimization accuracy and better improvement in the convergence performance.

2. The gravitational search algorithm

In the gravitational search algorithm, each particle is regarded as a mass moving object in space. It is attracted by particles by attracting gravitation. Particles with smaller mass move in the direction of mass particles. The motion of particles follows Newton's second law. With the continuous movement of motion, the largest mass of the particle occupies the best position in the end, corresponding to the better fitness value, and then the optimal solution of the problem is obtained.

Suppose there are *N* agents in the *D*-dimensional search space, then the position of the agent *i* is presented as Equation (1):

$$X_i = (x_i^1, x_i^2, ..., x_i^k, ..., x_i^d); i = 1, 2, ..., N \tag{1}$$

x_i^k represents the position of the agent *i* on the *k* dimension.

On the dimension *k*, the agent *i* is subjected to the force of the agent *j* at the time *t*:

$$F_{ij}^k(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}(x_i^k(t) - x_i^k(t)) \tag{2}$$

In Equation (2), $M_{aj}(t)$ is the inertial mass of the acting agent *j*. $M_{pi}(t)$ is the inertia mass of the acted agent *i*. ε is a very small constant. $G(t)$ is the gravitational constant at the time *t*, $R_{ij}(t)$ represents Euclidean distance of agents between X_i and X_j.

At the time of *t*, the sum of forces acting on the X_i on the *k* dimension is equal to the sum of the forces of all other agents.

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

$rand_j$ is a random number of uniform distribution between [0,1].

According to Newton's second law, the agent is driven by the gravitational force of other agents and produces acceleration. The calculation equation is represented below:

$$a_i^k(t) = F_i^k(t) / M_{ii}(t) \tag{4}$$

$M_{ii}(t)$ is the mass of the agent *i*.

In each iteration, the agent is calculated to get the acceleration to update the velocity and position of the agent. Update equation is presented as below:

$$v_i^k(t+1) = rand_i \times v_i^k(t) + a_i^k(t) \tag{5}$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \tag{6}$$

$rand_i$ is a random number of uniform distribution between [0,1].

The mass of inertia is based on the size of the adaptive value. The larger the mass of the agent is, the closer it is to the optimal value. According to the gravitational search algorithm, the inertia mass updating equation of agents presented as Equation (7), (8) and (9):

$$M_{ai} = M_{pi} = M_{ii} = M_i, i = 1, 2, .., N \tag{7}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{8}$$

$$M_i(t) = m_i(t) / \sum_{j=1}^{N} m_j(t) \tag{9}$$

$fit_i(t)$ is agent's fitness value , $best(t)$ is optimal solution at the time *t*, $worst(t)$ is worst solution. Calculation equation is represented below:

$$best(t) = \max_{i \in \{1,2,...,N\}} fit(t) \tag{10}$$

$$worst(t) = \min_{i \in \{1,2,...,N\}} fit(t) \tag{11}$$

3. MGSA

3.1. Adaptive trigger function of mutation

Through the analysis of the gravitational search algorithm, the movement of particles decreases continuously. Moreover, the particle lacks the mechanism to jump out of the local extremum in the later stage as the number of iterations increases, which results in low convergence accuracy in the solution of some problems. Therefore, this paper proposes adaptive trigger function of mutation to improve the algorithm. At the same time, the mutation trigger rate tends to decrease with the increase of iteration times. Because higher mutation rate can improve the diversity of the population, make the algorithm jump out of the local extremum, and provide the basis for searching for further depth in later stages. Also, decrease the mutation rate at the end of the algorithm to prevent the destruction of excellent mass at the beginning of the algorithm, so as to provide the possibility for better accuracy. The expression of adaptive trigger function of mutation is:

$$TR_i^t = e^{(-\frac{\dim}{2} \times \frac{t}{T})} \times (rand \times e^{(-\frac{i-1}{N})} + \varepsilon) \tag{12}$$

TR_i^t is the trigger function value of the *i*th particle at iteration number *t*, *t* is the current iteration, *T* is the total number of iterations, dim is the dimension, *rand* is a uniform random variable in the interval [0,1]. *N* is the population size, and ε is a constant and the value is in [0.25,0.35].The threshold value of the trigger function is 0.5. When $TR_i^t > 0.5$, the particle is mutated. After the simulation test, when ε is 0.3, about 60% of the masses in the initial stage of the algorithm mutate and the algorithm's optimization effect is better.

3.2. Mutation process

In the early stage of the algorithm, the number of particles that satisfy the mutation condition is large, and the method of uniform mutation [8] is used to move the mobile area in a larger stride. The particles that meet the mutation condition are close to the optimal solution area, if not, they move according to the original step to the best solution area. If so, in the early stage of the algorithm, the two modes of movement make particles close to the optimal solution region quickly. It can also improve the convergence rate and the probability of jumping out of the local extremum. The equation for updating particle position with uniform mutation is presented as below:

$$x_i(t) = (1+r)x_i(t), \quad rand \le (1-\frac{t}{T}) \tag{13}$$

r is a random number distributed uniformly between [0,1], and *rand* is a random number used to judge the mode of mutation between [0,1].

In the later period of the algorithm, the hybrid mutation of normal mutation and Laplace mutation meet the mutation conditions of individual particle combined. The Laplace distribution has the function characteristic of "sharp peak and thick tail". The sharp peak makes the area of Laplace distributed more near the expected value in the smaller area than the normal distribution. But the thick tail makes the Laplace distribution within 3 times the standard deviation of the expected value less than the normal distribution. Hybrid mutation makes a local area where the generated random value is close the center of the expected value as much as possible. In this way, the particles can mutate near the current optimal solution. It can maintain the appropriate mutation ability to jump out the local extremum at the later stage. It is also conducive to the deep search in the later stage of the algorithm and get better optimization accuracy.

In this paper, the normal variation function $Normal_i(t)$ based on the basis of normal distribution probability density function is presented as Equation (14):

$$Normal_i(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{(-\frac{(x_{best}-x_i^t)^2}{2\sigma^2})} \tag{14}$$

x_{best} is the current global optimal solution, σ is the standard deviation of normal distribution.

The Laplace mutation function $Laplace_i(t)$ based on the basis of Laplace distribution probability density function is presented as Equation (15):

$$Laplace_i(t) = \frac{1}{2\lambda} e^{(-\frac{|x_{best}-x_i^t|}{\lambda})} \tag{15}$$

$2\lambda^2$ is the variance of Laplace distribution.

Laplace normal hybrid mutation function $MLN_i(t)$ is presented as Equation (16):

$$MLN_i(t) = (1-\omega)Laplace_i(t) + \omega Normal_i(t) \tag{16}$$

ω is the hybrid weight parameter according to the experimental simulation test, the best value ω is 0.3.

x_{best} is the current global optimal solution. In the later period of the algorithm, its mass is larger and the gravitational force of other particles is greater, which makes the algorithm easy to fall into local extremum. At this time, the attraction of particles that can get out of x_{best} and jump out of the local extremum through the disturbance of the hybrid variation. The position update equation for hybrid mutation movement is presented as below:

$$x_i(t) = MLN_i(t)x_i(t), \quad rand > (1-\frac{t}{T}) \tag{17}$$

The combination of uniform mutation and hybrid mutation accelerates the speed of particles moving to the best solution area and improves the ability to jump out of the local extremum. On the other hand, it improves the optimization accuracy of the algorithm, so it improves the performance of algorithm optimization.

3.3. Algorithm steps

Step 1: Initialize the population randomly, initialize the position of the particle *X* and speed *V*.

Step 2: Make the population close to boundary, according to the objective function $f(x)$、 $x=(x_1,x_2,...,x_d)^k$ and get the optimum solution Equation (10),get the optimum solution and record the position of particle x_{best} .

Step 3: According to the Equation (12) the triggering function of mutation, determine whether each particle is variant or not. If it meets conditions of the mutation, the uniform mutation or Laplace-normal hybrid mutation operation is carried out according to the Equation (13) and (17). If not, the mutation doesn't happen.

Step 4: After updating population, according to the objective function, Equation (10) and (11), calculate the best value $best(t)$ and worst value $worst(t)$ of the current generations, and find out the position x_{best} , the current location of global optimal solution.

Step 5: Calculate the inertia mass of the particle *M* and is update the gravitational constant *G* according to the best value $best(t)$, the worst value $worst(t)$ obtained in the Step 4 and the Equation(9).

Step 6: Calculate the gravity of the surrounding particles to the individual according to the Equation (3).

Step 7: Calculate accelerated velocity of the particles by the Equation (4), and obtain the velocity of the particle by the Equation (5).

Step 8: Update the position of the particle according to the Equation (6).

Step 9: Back to the Step2 loop iteration until the cycle index or required precision is reached.

Step 10: End the cycle and output the result.

3.4. Convergence analysis of improved algorithm

The speed update Equation (5) and the position update Equation (6) reflect the global searching ability of the algorithm, which provides a basic basis for proving the convergence of the algorithm. Reference [9] and [10] use difference equations to analyze the convergence of bat algorithm and particle swarm algorithm respectively. This paper uses the same method to analyze the convergence of the improved gravitational search algorithm by establishing the difference equation.

The Equation (5) and Equation (6) show that all variables are multidimensional variables, but each one is independent. So, it can be simplified to one dimension to improve the algorithm.

In order to calculate easily, value of $rand_i$ and r is assumed as 0.5, the hybrid mutation function is regarded as the constant coefficient μ . So, the velocity Equation is simplified as Equation (18):

$$v(t+1)=\frac{1}{2}v(t)+a(t) \tag{18}$$

There are 3 forms of the position equation of the improved algorithm: position update equation for non-mutation individuals, the position update equation of uniform mutation individuals and position update equation of hybrid mutation individuals, as Equation (6), (19) and (20):

$$x(t+1)=1.5x(t)+v(t+1) \tag{19}$$

$$x(t+1)=\mu x(t)+v(t+1) \tag{20}$$

Case 1: Non mutation from Equation (5) and the Equation (6) can be obtained as shown in Equation (21):

$$x(t+2)\text{-}\frac{3}{2}x(t+1)+\frac{1}{2}x(t)=\mathrm{a(t+1)} \tag{21}$$

This is a second-order nonhomogeneous differential equation with constant coefficients and its characteristic equation is: $\lambda^2\text{-}\frac{3}{2}\lambda+\frac{1}{2}=0$. $\Delta=\frac{1}{4}>0$, $\lambda_{1,2}=\frac{\frac{3}{2}\pm\sqrt{\Delta}}{2}$, and $\lambda_1=1$, $\lambda_2=\frac{1}{2}$. Then, $x(t)=A_0+A_1\lambda_1^t+A_2\lambda_2^t$,and A_2 are

undetermined coefficient. $\lim_{t\to\infty} = A_0 + A_2$,the particle trajectories converge.

Case 2: Uniform mutation, from Equation (5) and the Equation (19), it can be obtained as Equation (22):

$$x(t+2)\text{-}2x(t+1)+\frac{3}{4}x(t) = \text{a(t+1)} \tag{22}$$

Its characteristic equation is $\lambda^2\text{-}2\lambda+\frac{3}{4}=0$. $\Delta = 1 > 0$, $\lambda_{1,2}=\frac{2\pm\sqrt{\Delta}}{2}$,and $\lambda_1=\frac{3}{2}$, $\lambda_2=\frac{1}{2}$. So similarly, the particle trajectory does not converge.

Case 3: Hybrid mutation, from Equation (5) and the Equation (20), it can be obtained as Equation (23):

$$x(t+2)\text{-}(\mu+\frac{1}{2})x(t+1)+\frac{1}{2}\mu x(t) = \text{a(t+1)} \tag{23}$$

Its characteristic equation is $\lambda^2\text{-}(\mu+\frac{1}{2})\lambda+\frac{1}{2}\mu=0$. $\Delta=\left(\mu+\frac{1}{2}\right)^2\text{-}2\mu$ that is $\Delta=\left(\mu\text{-}\frac{1}{2}\right)^2\geq 0$. There are 2 conditions to be considered:

1) When $\Delta = 0$, $\lambda_1=\lambda_2=\frac{\mu}{2}+\frac{1}{4}$,and $x(t)=(A_0+A_1t)\lambda^t$, A_0、A_1 are undetermined coefficients.

2) When $\Delta > 0$, $\lambda_{1,2}=\frac{(\mu+\frac{1}{2})\pm\sqrt{\Delta}}{2}$,and $x(t)=A_0+A_1\lambda_1^t+A_2\lambda_2^t$, A_0、A_1、A_2 are undetermined coefficients.

If $t\to\infty$, $x(t)$ has its maximum and tend to finite value. It is iterative convergence. According to above two situations, $x(t)$ conditions of convergence is $\|\lambda_1\|<1$ and $\|\lambda_2\|<1$.

It is calculated as below:

When $\Delta = 0$,the convergent domain is $\mu=\frac{1}{2}$.

When $\Delta>0$,the convergent domain is $\mu<\frac{3}{2}$ and $\mu\neq\frac{1}{2}$.

In conclusion, the condition of iterative convergence is $\mu<\frac{3}{2}$.

In the hybrid mutation, $\mu<\frac{3}{2}$, so the hybrid mutation is convergence.

In the whole iteration process of the algorithm, if the particle conforms to the mutation condition, according to Equation (13) and (17), the uniform mutation exists only in the early stage of the algorithm. The later stage is the hybrid mutation, so the algorithm is convergent when the particle changes. It can be seen that the improved algorithm converges.

4. Function optimization simulation and result analysis

In order to test the optimization performance of the improved algorithm (MGSA), this paper designs simulation experiment, selects bat algorithm (BA) [17], gravitational search algorithm (GSA), improved gravitation search algorithm(IGSA) [7] and MGSA to make contrast simulation through 8 classical test functions.

4.1. Optimization test function

These are 8 test functions in Table 1. $f_1(x)$ and $f_2(x)$ are unimodal functions and used to test the efficiency of the algorithm and the speed of convergence. $f_3(x)$, $f_4(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$ and $f_8(x)$ are multimodal functions and used to test the global search capability of the algorithm and the ability to jump out of local extremes and avoid precocity. $f_3(x)$

is a typical nonlinear multimodal function and the fluctuating peak is hopping. It is difficult to find the global optimal solution. Function $f_7(x)$ is a complex one of two dimensions, with numerous minimum points. The function is concussion and too difficult to find the global optimal solution. These 8 functions have some difficulties in solving the problem. It is suitable for the test algorithm to solve the optimization performance of the optimization problem of function extremum.

Table 1. Optimization test function.(Here,*D*: Dimension of the function.)

Name	Text function	f_{min}	*D*
Sphere	$f_1(x)=\sum_{i=1}^{n}x_i^2$	0	10,30,50
Schwefel's Problem 1.2	$f_2(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$	0	10,30,50
Generalized Rastrigin's	$f_3(x)=\sum_{i=1}^{n}(x_i-10\cos 2\pi x_i+10)$	0	10,30,50
Zakharov's	$f_4(x)=\sum_{i=1}^{n}x_i^2+(\sum_{i=1}^{n}\frac{i}{2}x_i)^2+(\sum_{i=1}^{n}\frac{i}{2}x_i)^4$	0	10,30,50
Alpine	$f_5(x)=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	0	10,30,50
Salomon	$f_6(x)=-\cos\left(2\pi\sqrt{\sum_{i=1}^{n}x_i^2}\right)+0.1\times\sqrt{\sum_{i=1}^{n}x_i^2}+1$	0	10,30,50
Schaffer	$f_7(x)=0.5+\frac{(\sin\sqrt{x_1^2+x_2^2})^2-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	0	2
Eggcrate	$f_8(x)=x_1^2+x_2^2+25(\sin^2 x_1+\sin^2 x_2)$	0	2

4.2. Analysis of optimization precision

In order to avoid errors caused by contingency and ensure objectivity and fairness in the experiment, four algorithms are run independently in the same environment for 30 times and the dimensions are10, 30, 50, respectively. The test dimension is set to 2 because the function $f_7(x)$ and $f_8(x)$ are two dimensional function. The parameter settings are consistent with the basic gravitational search algorithm, that is to say, the size of the population is set to N=50, the number of iterations is 1000, the gravitational constant G_0 is 100, and the value of α is 20. As for the bat algorithm, the loudness is 0.25 and the pulse rate is 0.5. In IGSA, the minimum weight of the inertia weight is 0.6 and the maximum weight is 0.9.

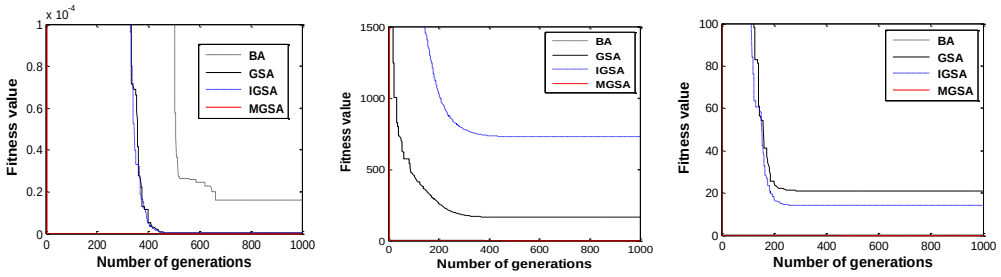
Table 2 show that under the same dimension, the optimization precision of MGSA is obviously higher than that of BA, GSA and IGSA. The globally optimal solution of MGSA is obtained under different dimensions. For the other three algorithms, only IGSA can get the globally optimal solution only in the $f_7(x)$. It can be seen that the optimization effect of the 4 functions has reached the best effect. As for the function $f_3(x)$, MGSA can achieve the global optimum on 30 and 50 dimensions, but there is no probability of reaching the optimal value in 10 dimensions. This is caused by fewer particles, while the other three algorithms cannot achieve global optimum in all dimensions. As for the function $f_5(x)$, MGSA has no advantages in the 10 dimensional optimizations; but, with the increase of the dimension, the precision is continuously improved. For multidimensional functions, BA, GSA, and IGSA decrease with the increase of dimensions, and even fall into local extremes. But MGSA prevents this condition. This is due to the adaptive mutation mechanism that makes the particles mutate. For the function $f_8(x)$, the four algorithms cannot find the global optimal value, but the degree of accuracy of the MGSA is obviously higher than that of the other three algorithms. The simulation results show that MGSA has wide adaptability whether in 10 dimensional, 30 and 50 dimensional multidimensional functions, or on 2 dimensional functions. This shows the obvious improvement of the convergence precision of the improved algorithm.

4.3. Analysis of convergence performance

The convergence curve of the algorithm directly reflects the number of local optimal solutions and the speed of downward convergence and it is an important index to measure the performance of the algorithm. There are 4 algorithm convergence curves . The dimension of functions $f_1(x), f_2(x), f_3(x), f_4(x), f_5(x), f_6(x)$ is 30. The dimension of functions $f_7(x)$ and $f_8(x)$ is 2. It can be seen in Figure 1 to Figure 3.

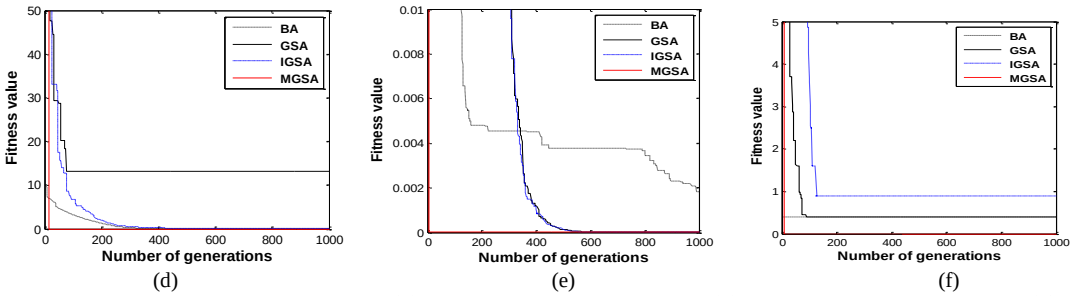
Table 2. 4 algorithms simulation results in test functions

Algorithm	Function	D	Worst	Best	Mean	Function	D	Worst	Best	Mean
BA	$f_1(x)$	10	1.109e-07	5.263e-07	8.208e-07	$f_1(x)$	30	1.890e-05	1.125e-05	1.435e-05
GSA			3.081e-18	5.765e-19	1.645e-18			2.740e-17	1.134e-17	1.903e-17
IGSA			8.170e-19	1.746e-19	4.882e-19			8.123e-18	2.422e-18	4.430e-18
MGSA			**0**	**0**	**0**			**0**	**0**	**0**
BA	$f_1(x)$	50	5.493e-05	3.731e-05	4.470e-05	$f_2(x)$	10	1.712e-06	6.269e-07	9.985e-07
GSA			9.932e-17	3.867e-17	6.622e-17			9.623e-18	1.493e-18	4.291e-18
IGSA			9.500e+02	3.305e-17	2.282e+02			2.731e-18	3.733e-19	1.220e-18
MGSA			**0**	**0**	**0**			**0**	**0**	**0**
BA	$f_2(x)$	30	2.769e-04	2.670e-05	1.023e-04	$f_2(x)$	50	1.514e+00	1.742e-03	1.548e-01
GSA			4.660e+02	5.577e+01	2.442e+02			1.805e+03	5.351e+02	1.040e+03
IGSA			2.366e+03	3.533e+02	1.146e+03			1.178e+04	3.220e+03	6.639e+03
MGSA			**0**	**0**	**0**			**0**	**0**	**0**
BA	$f_3(x)$	10	2.473e-12	1.776e-15	5.713e-13	$f_3(x)$	30	8.329e-12	0	1.344e-12
GSA			4.975	0.995	3.327			2.389e+01	6.965e+00	1.489e+03
IGSA			6.694	0.995	3.482			3.781e+01	5.970e+00	1.340e+01
MGSA			**0**	**0**	**0**			**0**	**0**	**0**
BA	$f_3(x)$	50	8.834e-12	3.553e-15	1.214e-12	$f_4(x)$	10	3.114e-06	6.181e-07	1.551e-06
GSA			4.378e+01	1.990e+01	3.028+01			9.990e-18	1.368e-18	4.641e-18
IGSA			3.184e+01	1.194e+01	2.161e+01			2.291e-18	7.355e-19	1.419e-18
MGSA			**0**	**0**	**0**			**5.322e-188**	**0**	**3.433e-189**
BA	$f_4(x)$	30	3.135e+02	4.455e-05	4.478e+01	$f_4(x)$	50	5.233e+02	8.138e+00	1.455e+02
GSA			2.295e+01	9.552e+00	1.547e+01			4.408e+01	2.031e+01	3.162e+01
IGSA			2.047e-01	5.076e-03	6.228e-02			7.433e+00	1.414e+00	3.579e+00
MGSA			**0**	**0**	**0**			**0**	**0**	**0**
BA	$f_5(x)$	10	2.690e-04	1.698e-04	2.147e-04	$f_5(x)$	30	3.946e-03	1.349e-03	2.059e-03
GSA			4.807e-10	1.226e-10	3.637e-10			3.051e-09	1.768e-09	2.352e-09
IGSA			2.508e-10	1.272e-10	2.033e-10			1.389e-09	7.032e-10	9.086e-10
MGSA			5.408e-10	2.175e-10	3.908e-10			**1.944e-13**	**2.153e-26**	**2.296e-14**
BA	$f_5(x)$	50	1.242e-02	3.704e-03	6.086e-03	$f_6(x)$	10	9.949e+00	8.955e-01	3.930e+00
GSA			8.698e-03	3.963e-09	1.091e-03			1.003e-01	9.950e-02	9.955e-02
IGSA			1.389e-02	9.374e-04	7.382e-03			9.950e-02	9.950e-02	9.950e-02
MGSA			**1.262e-19**	**2.149e-41**	**8.466e-21**			**0**	**0**	**0**
BA	$f_6(x)$	30	2.547e+01	6.368e+00	1.537e+01	$f_6(x)$	50	4.814e+01	9.949e+00	2.813e+01
GSA			7.508e-01	3.980e-01	4.203e-01			4.875e+00	8.955e-01	1.906e+00
IGSA			7.797e+01	2.487e+00	4.204e+01			8.363e+01	1.950e+01	4.615e+01
MGSA			**0**	**0**	**0**			**0**	**0**	**0**
BA	$f_7(x)$	2	3.722e-02	9.716e-03	1.315e-02	$f_8(x)$	2	1.898e+01	1.905e-11	2.965e+00
GSA			9.719e-03	2.312e-04	5.273e-03			6.880e-19	6.875e-22	1.370e-19
IGSA			0	0	0			4.207e-19	2.875e-21	1.106e-18
MGSA			**0**	**0**	**0**			**1.244e-29**	**5.716e-34**	**1.333e-30**



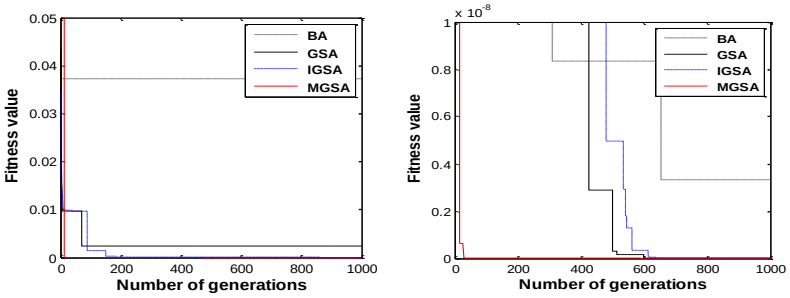
(a) (b) (c)

Figure 1. (a) the convergence curve of $f_1(x)$, (b) the convergence curve of $f_2(x)$, (c) the convergence curve of $f_3(x)$.



(d) (e) (f)

Figure 2. (d) the convergence curve of $f_4(x)$, (e) the convergence curve of $f_5(x)$, (f) the convergence curve of $f_6(x)$.



(g) (h)

Figure 3. (g) the convergence curve of $f_7(x)$, (h) the convergence curve of $f_8(x)$.

Figures 1-3 show that MGSA can achieve final convergence within 30 generations. The convergence rate is higher than other algorithms, and has higher convergence performance. Figure (a) and (e) show that the other three algorithms converge after 400 generations. Figure (d) shows that GSA converges around 100 generations and IGSA and BA converge at 400 generations. Figure (h) shows GSA, BA, and IGSA converge around 600 generation. In Figure (b) and Figure (c) show that GSA and IGSA, their precision is bad. Figure (f) shows BA, GSA and IGSA, their precision is low, in order to make the convergence curve of four algorithms can be shown in the same image, the target value is high and cannot compare the convergence properties of MGSA and BA. If the range is set more accurately, it can show that the convergence performance of MGSA is better than BA. Figure (g) shows that the convergence performance of MGSA is very close to BA and they are better than GSA and IGSA while the accuracy of MGSA is obviously higher than that of BA. Analysis of the convergence curves of 4 kinds of algorithms, the convergence curve of MGSA is almost vertical, frequency of arc and ladder form was significantly less than the other three algorithms. The reason is that the algorithm can quickly converge to the optimal solution area by using uniform mutation in the early stage and in the later stage, the Laplace - normal hybrid mutation is used. It can jump out of the local extremum and near the optimal solution, and then quickly converge to the optimal solution. The simulation results show that in terms of function optimization, the improved algorithm in this paper (MGSA), its convergence speed is faster than other three algorithms. MGSA also still maintains higher efficiency of finding optimal solution in the treatment of the multi-dimensional and multi-extremal problems.

As a whole, this algorithm (MGSA) has fewer iterations, faster speed of convergence, better precision of convergence and better performance of optimization.

5. Conclusions

Aiming at shortcomings of the gravitational search algorithm such as bad precision and easy to fall into the local optimal solution, this paper introduces and constructs adaptive trigger function of mutation. This kind of function generates mutation individuals in the group and the number is random and has tendency. In the process of mutation, the strategy of combining uniform mutation and Laplace-normal mutation is applied, which improves the algorithm's ability of jumping out of the local extremum and also better solves the problems such as slow location of the optimal solution at previous stage, low precision of optimization at the later stage. The simulation results show that the optimization precision of MGSA is greatly improved and has good performance in convergence performance.

References

1. H. Askari, S. H. Zahiri. “Decision Function Estimation Using Intelligent Gravitational Search Algorithm,” *International journal of machine learning and cybernetics*, vol. 3, no. 2, pp. 163-172, 2012.
2. A. Bahrololoum, H. Nezamabadi-Pour, H. Bahrololoum, et a1. “A Prototype Classifier Based on Gravitational Search Algorithm,” *Applied Soft Computing*, vol. 12, no. 2, pp. 819-825, 2012.
3. B. Gu, F. Pan. “Modified Gravitational Search Algorithm with Particle Memory Ability and its Application,” *International Journal of Innovative Computing Information & Control*, vol. 9, no. 11, pp. 4531-4544, 2013.
4. M. Khajehzadeh, M. R. Taha, A. E1-Shafie, et a1. “A Modified Gravitational Search Algorithm for Slope Stability Analysis,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1589-1597, 2012.
5. J. V. Kumar, D. M. V. Kumar, and K. Edukondalu. “Strategic Bidding Using Fuzzy Adaptive Gravitational Search Algorithm in a Pool Based Electricity Market,” *Applied Soft Computing*, vol. 13, no. 5, pp. 2445-2455, 2013.
6. C. Li, J. Zhou, and J. Xiao, et a1. "Parameters Identification of Chaotic System by Chaotic Gravitational Search Algorithm," *Chaos Solitons & Fractals,* vol. 45, no. 4, pp. 539-547, 2012.
7. P. Li, H. B. Duan. “Path Planning of Unmanned Aerial Vehicle Based on Improved Gravitational Search Algorithm,” *SCIENTIA SINICA Technologica,* vol. 55, no. 10, pp. 2712-2719, 2012.
8. Y. Li, Y. H. Pei, J. S. Liu. “Bat Optimal Algorithm Combined Uniform Mutation with Gaussian Mutation,” *Control and*

Decision, vol. 32, no. 10, 2017.
9. Z. Y. Li, L. Ma, and H. Z. Zhang. "Convergence Analysis of Bat Algorithm," *Journal of Mathematics in Practice and Theory*, vol. 43, no. 12, pp. 182-190, 2013.
10. H. B. Liu, X. K. Wang, and G. Z. Tan. "Convergence Analysis of Particle Swarm Optimization and Its Improved Algorithm Based on Chaos," *Control and Decision*, vol. 21, no. 6, pp. 636-640, 2006.
11. S. Mirjalili, A. Lewis. "Adaptive Gbest-guided Gravitational Search Algorithm," *Neural Computing & Applications*, vol. 25, no. 7-8, pp. 1569-1584, 2014.
12. S. Mondal, A. Bhattacharya. "Multi-objective Economic Emission Load Dispatch Solution Using Gravitational Search Algorithm and Considering Wind Power Penetration," *International Journal of Electrical Power &Energy Systems*, vol. 44, no. 1, pp. 282-292, 2013.
13. E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi. "Filter Modeling Using Gravitational Search Algorithm," *Engineering Applications of Artificial* Intell*igence*, vol. 24, no. 1, pp. 117-122, 2011.
14. E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi. "GSA: A Gravitational Search Algorithm," *Information sciences*, vol. 179, no. 13, pp. 2232-2248, 2009.
15. P. Song, Y. He, and Q. Ma. Song, Ping, Y. He, and Q. Ma. "Fault Diagnosis for Missile Autopilot Based on GSA-SVM," *Advanced Information Management, Communicates, Electronic and Automation Control Conference* IEEE, pp. 1365-1369, 2017.
16. G. Sun, A. Zhang, X. Jia, et al. "DMMOGSA: Diversity-enhanced and Eemory-based Multi-objective Gravitational Search Algorithm," *Information Sciences*, vol. 365, pp. 52-71, 2016.
17. X. S. Yang. "A New Metaheuristic Bat-Inspired Algorithm," *Computer Knowledge & Technology*, vol. 284, pp. 65-74, 2010.
18. B. Zibanezhad, K. Zamanifar, and R. S. Sadjady, et a1. "Applying Gravitational Search Algorithm in the QoS-based Web Service Selection Problem," *Journal of Zhejiang University SCIENCEC*, vol. 12, no. 9, pp. 730-742, 2011.

Jingsen Liu received his PhD from Northwestern Polytechnical University. He is a professor of Institute of Intelligent Network system, and College of Software, Henan University. His research interests include intelligence algorithm and network information security, etc.

Yuhao Xing is a master degree candidate of College of Software, Henan University. His research interest is intelligence algorithm.

Yu Li received her PhD from University of Shanghai for Science and Technology. She is a professor of Institute of Management Science and Engineering, Henan University. Her research interests include intelligence algorithms, electronic commerce, etc.