Gravitational Search Algorithm Based on Simulated Annealing

TONG Chengyi
Department of Electronics and Information, Changsha Normal University, Changsha, Hunan, China, Postcode: 410100, Email: 329699077@qq.com

Abstract

Because the individual position update strategy in Gravitational Search Algorithm (GSA) may cause damage to the individual position and the local search ability of GSA is weak, this paper proposed an improved algorithm. The new algorithm introduced the idea of Simulated Annealing into GSA, took the Metropolis-principle-based individual position update strategy to improve the particle moves, and after the operation of gravitation, applied the Simulated Annealing operation to the optimal individual. The experimental results show: the improved strategies of the new algorithm are effective to overcome the randomness of the individual moves and to enhance the local search ability of the algorithm. The improved algorithm has obvious advantages in convergence speed, convergence accuracy and so on.

Key words: *Gravitational Search Algorithm ; Simulated Annealing ; Function Optimization ; GSA*

1. Introduction

In science, economics and engineering fields, many of the latest developments are dependent on global optimization technique, that is calculates numerical techniques for global optimization problems corresponding to the optimal solution [1]. Heuristic Search Algorithm is one of the most effective global optimization techniques. At present, comparatively more commonly used heuristic search algorithms are Particle Swarm Optimization Algorithm[2], Simulated Annealing Algorithm[3], Genetic Algorithms [4],Electromagnetism-like Mechanism Algorithm [5], Ant Colony Optimization [6], and Gravitational Search Algorithm [7], etc. Despite such algorithm provides many effective solutions for solving complex optimization problems, its results are still unsatisfactory, for example, the algorithm is easy to fall into local optimal solution and low accuracy. On the other hand, most heuristic algorithms still have no systematic theories, no uniform algorithm framework, but many issues to be investigated [8].

Gravitational Search Algorithm (GSA) is proposed by Professor Esmat Rashedi of Blackman University of Iranian in 2009. The algorithm is based on Newton's law of universal gravitation, and directs the particles in search through simulated particle-particle interaction. Text [7] found that compared with Central Force Algorithm and Particle Swarm Optimization Algorithm, Gravitational Search Algorithm takes obvious advantages. However, the algorithm has certain randomness in the particle-move step resulting in individuals prone to particle degradation. The algorithm without local search mechanism is weak in local search ability which thus affects the convergence speed and accuracy.

According to the above mentioned disadvantages of GSA, this paper proposed a Gravitational Algorithm Based Simulated Annealing (GABSA). The new algorithm uses the Simulated Annealing Algorithm Metropolis principle for comparing the positions of moving particles to control the particle moves in order to overcome their randomness; at the same time, it improves the local search ability through the use of Simulated Annealing Algorithm for local search. The experimental results showed that the optimization effect of GABSA is significantly better than that of GSA.

2. Simulated Annealing Algorithm

Simulated Annealing (SA) was firstly proposed by Metropolis in 1953. Ever since Kirkpatrick applied it to combinatorial optimization problems in 1983, it has successfully solved Vehicle Routing Optimization, TSP problem, Workshop Scheduling Optimization problems, etc [9-11].

Simulated Annealing Algorithm is derived from the Solid Annealing Ideological Theory: the solid is heated to a sufficiently high temperature, then left to cool down slowly. When heated, with temperature increasing, solid particles turn into disordered state and the internal energy increases; later when slowly cool down, particles gradually go back to order, reach equilibrium at each temperature level, and finally reach the ground state at normal temperature, the internal energy reduced to a minimum. Based on the similarity of the annealing process and general combinatorial optimization problems, Simulated Annealing Algorithm simulated internal energy E as objective function value f, temperature T as control parameter t, then repeated iteration as t gradually decay of values thus obtained the approximate optimal solution.

Simulated Annealing Algorithm started from a higher initial temperature, then as the temperature parameters decline, combined with random saltation in the solution space to randomly search the global optimal solution of the objective function, that is, the optimal solution can jump out at local and eventually become a global optimum.

Metropolis principle:

Let the initial state of the particle *i*, random perturbation to get a new state of the particle *j*. *E (i), E (j)* respectively the energy in the state of particles *i, j* .

(1) if E(j)<E(i)，The state transition is accepted；

(2) if E (j) ≥E (í) ，The state transition probability is:

$$P_{ij} = \exp(-\frac{E(j) - E(i)}{KT})$$

Where *K* is the physics terms of Boltzmann constant, *T* is the temperature of the material.

The basic steps of Simulated Annealing Algorithm see the article [12].

3．Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) is proposed by Iranian Professor Esmat Rashedi in 2009. The algorithm is based on the Newton’s law of gravitation: “Any two particles in the universe attract each other with a force in the direction of the center line. The force is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. The chemical nature of the material or physical state and intermediary substance has no links with the gravity.”

The formula could be expressed as:

$$F = G\frac{M_1 M_2}{R^2} \tag{1}$$

Where *F* represents the gravity, *G* represents the gravitational constant, M_1 and M_2 respectively represent the inertial mass of the two particles, and *R* is the Euclid Distance between two particles.

The attractiveness of particle *j* to particle *i* at dimension *d* is [7]:

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

Because the experimental result of *R* is better than that of R^2, use *R* instead of using R^2; M_{aj} represents the mass of exert force particle *j*, M_{pi} represents the mass of stressed particle *i*; ε is a small constant, *G* (t) is the gravitational constant at time *t*:

$$G(t) = G_0 e^{-\alpha\frac{t}{T}} \tag{3}$$

Where G_0 is the initial gravitational constant, α is a constant, *T* is the total number of iterations of the algorithm.

Pool efforts of particle *i* at dimension *d*:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand \times F_{ij}^d(t) \tag{4}$$

Here *rand* is a random quantity added to increase the randomness of the search algorithm, which ranges [0, 1].

According to the Newton's second law: at time *t*, the acceleration of particle *i* on d:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{5}$$

Here M_i *(t)* is the inertial mass of particle *i*. In GSA, the following formulas are used to update the inertial mass of the particle:

$$M_{ai} = M_{pi} = M_{ii} = M_i, (i = 1, 2, 3, \cdots, N) \tag{6}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{7}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{8}$$

Here *fit (t)* is the fitness value of particle. For the minimum problem, *worst (t)* and *best (t)* are defined as follow:

$$\text{best}(t) = \min_{j \in \{1, \cdots N\}} fit_j(t) \tag{9}$$

$$worst(t) = \max_{j \in \{1, \cdots, N\}} fit_j(t) \tag{10}$$

In GSA, every try of iteration will update particles' motion state in accordance with Newton's laws of motion. The formula is:

$$v_i^d(t+1) = rand \times v_i^d(t) + a_i^d(t) \tag{11}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{12}$$

Where *rand* is a random number from [0, 1], $v_i^d(t)$ and $x_i^d(t)$ respectively represents the speed and position proportion at dimension *d* of particle *i* at time *t*.

Find the specific steps of Gravitational Search Algorithm in article [7].

4 . Algorithm Improvement Based on Simulated Annealing

4.1. Position update strategy based on Metropolis principle

Formula (12) shows that the updating of particle position is to some degree random, so the individual may move from higher-fitness-value position to lower-fitness-value position. This phenomenon is called individual degradation, i.e. this may cause $fit_{i+1}(t) > fit_i(t)$ (for the minimum problem), which is obviously detrimental for solving the problem. To make up for the defect of gravitational algorithm, this paper, based on the Metropolis principle, proposed new strategies of particle position update: first use formula (12) to calculate the next position of particle *i* $\bar{x}_i(t+1)$; then use Metropolis principle to determine whether to accept $\bar{x}_i(t+1)$ as the next position of *i*. Details as follow:

If $fit_{i+1}(t) \le fit_i(t)$ or $rand \le \exp(-\frac{fit_{i+1}(t) - fit_i(t)}{KT})$,

$x_i(t+1) = \bar{x}_i(t+1)$;

else $x_i(t+1) = x_i(t)$.

Where $fit_{i+1}(t)$ is the fitness value of next possible position, $\bar{x}_i(t+1)$ of *i*, rand is a random parameter from [0, 1].

Where $fit_{i+1}(t)$ is the fitness value when $\bar{x}_i(t+1)$ is assumed to be the next possible position of *i*, rand is a random parameter from [0, 1].

It is shown that when the particles move from a higher-fitness-value position to a lower-fitness value position, if worse solution is accepted only at a certain probability with the guidance of Metropolis principle, the degradation of the particles could be avoided to some extent.

4.2. Gravitational Search Algorithm Based on Simulated Annealing

From article [7] we could see Gravitational Search Algorithm has strong global search ability; however, from the concrete steps of this algorithm we find it lack of local search mechanism. Annealing Algorithm, though not excellent in global searching at the early stage, can quickly search the local optimal solution[13]. Therefore, this article combined Gravitational Search Algorithm with Annealing Algorithm to render the algorithm both the global and local search capabilities.

The Gravitational Search Algorithm Based on Simulated Annealing complied with the basic frame of GSA [7], adopted the particle position update strategy based on Metropolis principle, and then when the gravitational algorithm is complete, use the optimal annealing to operate the optimal individual to increase the ability of local optimization algorithms. Specific steps of the algorithm are as follow:

(1) Initialization: including the initial index Q, initial gravitational constant G_0, the initial annealing temperature T_0, Boltzmann constant K, iterations N, etc.

(2) Calculate the particle fitness value.

(3) Update gravitational Parameter G (t), best (t), worst (t), the inertial mass of the particle Mi (t).

(4) Use formula (4) to calculate the sum of the force in each direction.

(5) Calculate the acceleration and velocity of each particle respectively according to formula (5) and (11).

(6) Calculate the next possible position of each particle according to formula (12). Use position update strategies in Metropolis principle to update the particle position.

(7) Operate the annealing of optimal Individuals.

(8) Determine the end of the iteration. If not end, then return to (3) and repeat.

(9) Output the result.

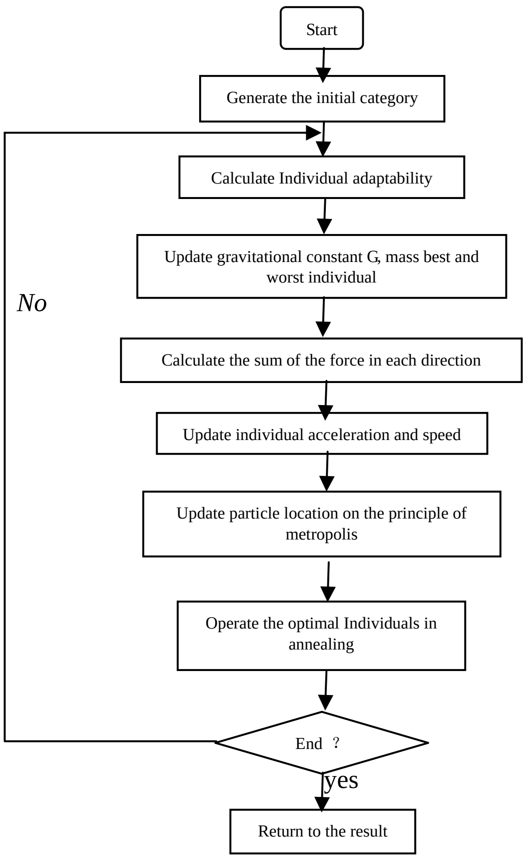

Fig 1. The Flowchart of GABSA

5． Algorithm Performance Tests

5.1． Performance test 1

In order to test the effectiveness of the participation of annealing, compare the new algorithm with GSA from article [7]. Table 1 is the test function [14]; the max iteration times of F_1 and F_2 is 50, and that of F_3 is 100; the optimize accuracy is 0.1. For other parameters please refer to table 2. Table3 demonstrates the comparison between this algorithm and GSA in article [7]. Fig 2-4 is the effect comparison of function F_1-F_3.

Table 1. Test Function 1

Function	Rang	Optimal value
$F_1\left(x_1,x_2\right)=x_1^2+x_2^2$	$x_i \in$ [-5,5]	$F_1\left(0,0\right)=0$
$F_2\left(x_1,x_2\right)=100(x_1^2-x_2)^2+\left(1-x_1\right)^2$	$x_i \in$ [-2.048,2.048]	$F_2\left(1,1\right)=0$
$F_3\left(x_1,x_2\right)=0.5+\dfrac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{\left[1.0+0.001(x_1^2+x_2^2)\right]^2}$	$x_i \in$ [-100,100]	$F_3\left(0,0\right)=0$

Table 2. Algorithm Parameters

parameters	Q	G0	α	T0	TM
GABSA	10	100	20	20	0.01
GSA	10	100	20	-	-

Table 3. Performance Comparisons between GABSA and GSA

function	GSA			GABSA		
	Success rate	Optimal solution	worst solution	Success rate	Optimal solution	worst solution
F_1	100%	1.911e-4	2.446e-3	100%	1.552e-6	2.066e-4
F_2	100%	1.445e-4	8.599e-2	100%	1.217e-4	1.694e-2
F_3	90%	1.744e-4	1.269e-1	100%	2.576e-6	9.720e-2

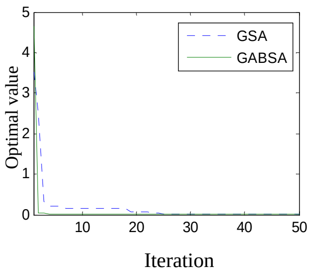

Fig 2. F_1 Optimizing curve

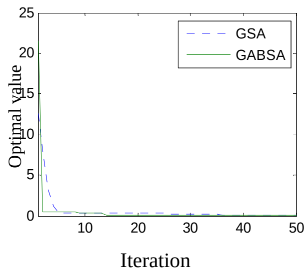

Fig 3. F2 Optimizing curve

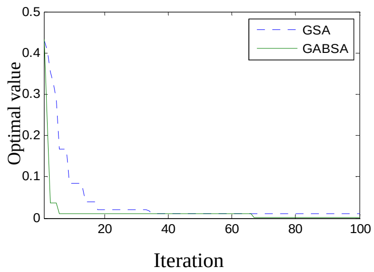

Fig 4.F_3 Optimizing curve

From table 3 we could see GABSA is better then GSA in success rate and solution accuracy. Fig 2-4 show GABSA is better than GSA in convergence speed and accuracy, because the employ of annealing algorithm makes the higher speed and accuracy. Firstly, individual reposition strategies in Metropolis principle avoid the individuals to move to worse solution; secondly, annealing operation on the optimal individual improved the convergence speed and search accuracy.

5.2. Performance test 2

To further test the performance of the algorithm, compare the new algorithm with species migration based optimization algorithm (SMOA) in article [15]. When testing, select high dimensional function from article [15] as test function(see table 4). Table 5 demonstrates the results, among which the SMOA test data is selected from article [15]. Experimental parameters: group scale is 30, evolution algebra is 50, and times of independent calculation are 50.

Table 4. Test Function 2

Function	Rang	Optimal value
$F_4(x) = \frac{1}{4000}\sum_{i=1}^{20} x_i^2 - \prod_{i=1}^{20}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in$ [-600,600]	$F_4(0,\cdots,0) = 0$
$F_5 = \sum_{i=1}^{20}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$x_i \in$ [-10,10]	$F_5(0,\cdots,0) = 0$
$F_6(x) = \sum_{i=1}^{19}\left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right]$	$x_i \in$ [-2,2]	$F_6(1,\cdots,1) = 0$
$F_7(x) = \sum_{i=1}^{20} x_i^2$	$x_i \in$ [-5.12,5.12]	$F_7(0,\cdots,0) = 0$

Table 5 .Performance Comparison between GABSA and SMOA

function	GABSA		SMOA	
	Optimal value	mean value	Optimal value	mean value
F4	0.2203	0.8738	6.7806	19.227
F5	2.9807	10.9723	43.517	54.484
F6	16.2365	21.7689	137.52	212.34
F7	6.8405e-3	2.6833e-2	2.4013	4.7472

As illustrated in Table 5, when solving high -dimensional function, GABSA is better than SMOA in convergence effect, solution accuracy, and optimization ability.

6 . Conclusion

According to the deficiencies of GSA, through the control of particle moves by individual reposition strategy based on Metropolis principle, and through the annealing operation on optimal individuals, this paper proposed a Gravitational Algorithm Based Simulated Annealing (GABSA). The experimental results showed that: GABSA is significantly effective; the improved algorithm has obvious advantages in convergence speed, convergence precision and so on.

References

[1] WANG Xiao-juan, GAO Liang, CHEN Ya-zhou, Electromagnetism-like Mechanism with its Application. Application Research of Computers, 23(6):67-70, 2006. (in Chinese)

[2] J Karakuzu, R C Eberhart.Particle swarm optimization.in: Proceedings of IEEE international conference on Neural Networks, 4:1942-1948,1995.

[3] Suman B,Kum ar P.A survey of simulated annealing as tool for single and multi-objective optimization.Journal of the Operational Research Society, 57 (10):1143-1160,2006.

[4] K S Tang, K F Man, S Kwong, Q He.Genetic algorithms and their applications.IEEE signal Processing Magazine, 13(6):22-37,1996.

[5] Birbil S I,Fang S C.An electromagnetism-like mechanism for global optimization.Journal of Global Optimozation, 25(3):263-282,2003.

[6] Dorigo M ,Birattari M,Stutzle T.Ant colony optimization-Artificial ants as a computational intelligence technique.IEEE Computational Intelligence Magazine, 1 (4):28-39,2006.

[7] Esmat Rashedi, Hossein Nezamabadipour, Saeid Saryazdi.GSA: A Gravitational Search Algorithm. Information Sciences, 179(13):2232-2248,2009.

[8] XIE Li-ping, ZENG Jian-chao. Physicomimetics-inspired global optimization algorithm. System Engineering Theory and Practice, 30(12):2276-2282,2010.

[9] Suman B, Kumar P. A survey of simulated annealing as a tool for single and multi-objective optimization. Journal of the Operational Research Society, 57 (10):1143-1160,2006.

[10] El-Bouri A, Azizi N, Zolfaghari S.A comparative study of a new heuristic based on fitness memory programming and simulated annealing: the case of job shop scheduling.European Journal of Operational Research, 177:1894-1910,2007.

[11] Tan K C, Chew Y H, Lee L H.A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems.European Journal of Operational Research, 172:855-885,2006.

[12] YANG Yong, TAN Yuan, ZHANG Xiao-fa, YUAN Nai-chang. Calibration of Array Model with Multiple Sources Based on Simulated Annealing. Journal of National University of Defense Technology, 33(1):91-94,2011. (In Chinese)

[13] WANG Yin-nian, GE Hong-wei. Improved simulated annealing genetic algorithm for solving TSP problem. Computer Engineering and Applications, 46(5):44-85,2010. (in Chinese)

[14] WU Tao, JIN Yi-fu. Adaptive Genetic Algorithm Based on Cloud Control. Computer Engineering, 37(8):189-191,2011. (in Chinese)

[15] MA Hai-ping, LI Huan, RUAN Xie-yong. Species migration-based optimization algorithm and performance analysis. Control Theory & Applications, 27(3):329-33,2010. (in Chinese)