

Article

Function Optimization and Parameter Performance Analysis Based on Gravitation Search Algorithm

Jie-Sheng Wang ^{1,2,*} and Jiang-Di Song ¹

Received: 10 October 2015; Accepted: 21 December 2015; Published: 24 December 2015

Academic Editor: Paul M. Goggans

¹ School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114044, China; sjd2011@163.com

² National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning, Anshan 114044, China

* Correspondence: wang_jiesheng@126.com or wangjiesheng@ustl.edu.cn; Tel.: +86-412-5929699

Abstract: The gravitational search algorithm (GSA) is a kind of swarm intelligence optimization algorithm based on the law of gravitation. The parameter initialization of all swarm intelligence optimization algorithms has an important influence on the global optimization ability. Seen from the basic principle of GSA, the convergence rate of GSA is determined by the gravitational constant and the acceleration of the particles. The optimization performances on six typical test functions are verified by the simulation experiments. The simulation results show that the convergence speed of the GSA algorithm is relatively sensitive to the setting of the algorithm parameters, and the GSA parameter can be used flexibly to improve the algorithm's convergence velocity and improve the accuracy of the solutions.

Keywords: gravitational search algorithm; function optimization; performance comparison

1. Introduction

The function optimization problem is to find the optimal solution of the objective function by the iterative [1]. In general, the search objective is to optimize the function of the objective function, which is usually described by the continuous, discrete, linear, nonlinear, concave and convex of the function. For constrained optimization problems, one can adopt the specific operators to make the solutions always feasible, or use the penalty function to transform the solutions into the unconstrained problems. So, the unconstrained optimization problems are the research focus here. There has been considerable attention paid to employing metaheuristic algorithms inspired from natural processes and/or events in order to solve function optimization problems. The swarm intelligent optimization algorithm [2] is a random search algorithm simulating the evolution of biological populations. It can solve complex global optimization problems through the cooperation and competition among individuals. The representative swarm intelligence optimization algorithms include Ant Colony Optimization (ACO) algorithm [3], Genetic Algorithm (GA) [4], Particle Swarm Optimization (PSO) algorithm [5], Artificial Bee Colony (ABC) algorithm [6], etc.

However, not all met heuristic algorithms are bio-inspired, because their sources of inspiration often come from physics and chemistry. For the algorithms that are not bio-inspired, most have been developed by mimicking certain physical and/or chemical laws, including electrical charges, gravity, river systems, etc. The typical physics and chemistry inspired met heuristic algorithms include Big Bang-big Crunch optimization algorithm [7], Black hole algorithm [8], Central force optimization algorithm [9], Charged system search algorithm [10], Electro-magnetism optimization algorithm [11], Galaxy-based search algorithm [12], Harmony search algorithm [13], Intelligent water

drop algorithm [14], River formation dynamics algorithm [15], Self-propelled particles algorithm [16], Spiral optimization algorithm [17], Water cycle algorithm [18], *etc.*

The gravitational search algorithm (GSA) was introduced by E. Rashedi *et al.* in 2009 [19]. It was constructed based on the law of gravity and the notion of mass interactions. The GSA algorithm uses the theory of Newtonian physics and its searcher agents are the collection of masses. A multi-objective gravitational search algorithm (MOGSA) technique was proposed to be applied in hybrid laminates to achieve minimum weight and cost [20]. A fuzzy gravitational search algorithm was proposed for a pattern recognition application, which in addition provided a comparison with the original gravitational approach [21]. The grouping GSA (GGSA) adapted the structure of GSA for solving the data clustering problem [22]. Combining the advantages of the gravitational search algorithm (GSA) and gauss pseudo spectral method (GPM), an improved GSA (IGSA) is presented to enhance the convergence speed and the global search ability [23]. A novel modified hybrid Particle Swarm Optimization (PSO) and GSA based on fuzzy logic (FL) was proposed to control the ability to search for the global optimum and increase the performance of the hybrid PSO-GSA [24]. A binary quantum-inspired gravitational search algorithm (BQIGSA) was proposed by using the principles of QC together with the main structure of GSA to present a robust optimization tool to solve binary encoded problems [25]. Another binary version of hybrid PSOGSA called BPSOGSA was proposed to solve many kinds of optimization problems [26]. A new gravitational search algorithm was proposed to solve the unit commitment (UC) problem, which is the integrated binary gravitational search algorithm (BGSA) with the Lambda-iteration method [27].

In conclusion, GSA has been successfully applied in many global optimization problems, such as, multi-objective optimization of synthesis gas production [28], the forecasting of turbine heat rate [29], dynamic constrained optimization with offspring repair [30], fuzzy control system [31], grey nonlinear constrained programming problem [32], reactive power dispatch of power systems [33], minimum ratio traveling salesman problem [34], parameter identification of AVR system [35], strategic bidding [36], *etc.* In this paper, the research on the function optimization problem is solved based on the gravitation search algorithm (GSA). Then the parameter performance comparison and analysis are carried out through the simulation experiments in order to verify its superiority. The paper is organized as follows. In Section 2, the gravitational search algorithm is introduced. The simulation experiments and results analysis are introduced in details in Section 3. Finally, the conclusion illustrates the last part.

2. Gravitational Search Algorithm

2.1. Physics Foundation of GSA

The law of universal gravitation is one of the four basic forces in nature. It is one of the fundamental forces in nature. The gravitational force is proportional to the product of the mass, and is inversely proportional to the square of the distance. The gravitational force between two objects is calculated by:

$$F = G \frac{M_1 \cdot M_2}{R^2} \quad (1)$$

where, F is the gravitational force between two objects, G is the gravitational constant, M_1 and M_2 are the masses of the objects 1 and 2 respectively, R is the distance between these two objects. According to the international unit system, the unit of F is Newton (N), the unit of M_1 and M_2 is kg, the unit of R is m, and the constant G is approximately equal to $6.67 \times 10^{-11} \text{ N} \cdot \text{m}^2/\text{kg}^2$.

The acceleration of the particle a is related to its mass M and of the gravitational force F , which is calculated by the following equation.

$$a = \frac{F}{M} \quad (2)$$

According to the Equations (1) and (2), all of the particles in the world are affected by gravity. The closer the distance between two particles, the greater the gravitational force. Its basic principle

is shown in Figure 1, where the mass of the particles is represented by the image size. Particle M_1 is influenced by the gravity of the other three particles to produce the resultant force F . Such an algorithm will converge to the optimal solution, and the gravitational force will not be affected by the environment, so the gravity has a strong local value.

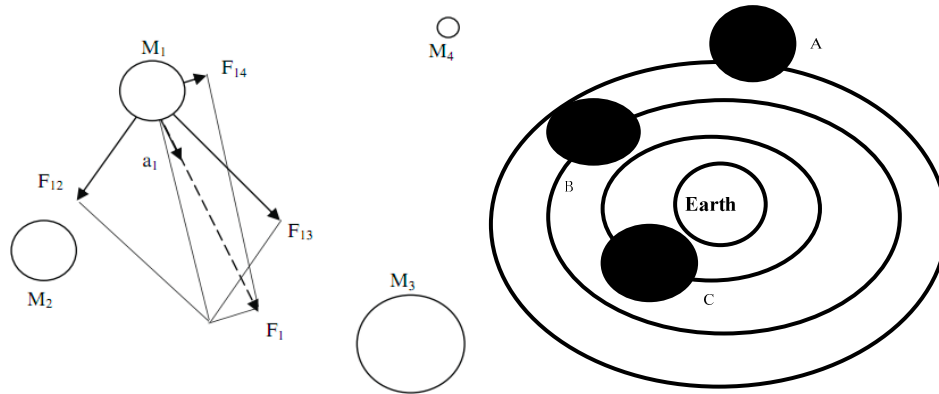


Figure 1. Gravitational phenomena.

The gravitational search algorithm should make the moving particle in space into an object with a certain mass. These objects are attracted through gravitational interaction between each other, and each particle in the space will be attracted by the mutual attraction of particles to produce accelerations. Each particle is attracted by the other particles and moves in the direction of the force. The particles with small mass move to the particles with great mass, so the optimal solution is obtained by using the large particles. The gravitation search algorithm can realize the information transmission through the interaction between particles.

2.2. Basic Principles of Gravitation Search Algorithm

Because there is no need to consider the environmental impact, the position of a particle is initialized as X_i . Then in the case of the gravitational interaction between the particles, the gravitational and inertial forces are calculated. This involves continuously updating the location of the objects and obtaining the optimal value based on the above mentioned algorithm. The basic principal of gravitational search algorithm is described as follows in detail.

2.2.1. Initialize the Locations

Firstly, randomly generate the positions $x_i^1, x_i^2, \dots, x_i^k, \dots, x_i^d$ of N objects, and then the positions of N objects are brought into the function, where the position of the i th object are defined as follows.

$$X_i = (x_i^1, x_i^2, \dots, x_i^k, \dots, x_i^d) \quad (3)$$

2.2.2. Calculate the Inertia Mass

Each particle with certain mass has inertia. The greater the mass, the greater the inertia. The inertia mass of the particles is related to the self-adaptation degree according to its position. So the inertia mass can be calculated according to the self-adaptation degree. The bigger the inertial mass, the greater the attraction. This point means that the optimal solution can be obtained. At the moment t , the mass of the particle X_i is represented as $M_i(t)$. Mass $M_i(t)$ can be calculated by the followed equation.

$$M_{ai} = M_{pi} = M_{ii} = M_i \quad (4)$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (5)$$

$$M_i(t) = \frac{m_i(t)}{\sum_N m_j(t)} \quad (6)$$

where $i = 1, 2, \dots, N$, $fit(i)$ is the fitness value of the object i , $best(t)$ is the optimal solution and $worst(t)$ is the worst solution. The calculation equation of $best(t)$ and $worst(t)$ are described as follows.

For solving the maximum problem:

$$best(t) = \max_{j=\{1,2,\dots,N\}} fit_j(t) \quad (7)$$

$$worst(t) = \min_{j=\{1,2,\dots,N\}} fit_j(t) \quad (8)$$

For solving the minimum value problem:

$$best(t) = \min_{j=\{1,2,\dots,N\}} fit_j(t) \quad (9)$$

$$worst(t) = \min_{j=\{1,2,\dots,N\}} fit_j(t) \quad (10)$$

2.2.3. Calculate Gravitational Force

At the moment t , the calculation formula for the gravitational force of object j to object i described as follows.

$$F_{ij}^k = G(t) \frac{M_{pi}(t) \cdot M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^k(t) - x_i^k(t)) \quad (11)$$

where, ε is a very small constant, $M_{ai}(t)$ is the inertial mass of the object itself, $M_{pi}(t)$ is the inertial mass of an object i . $G(t)$ is the universal gravitational constant at the moment t , which is determined by the age of the universe. The greater the age of the universe, the smaller $G(t)$. The inner relationship is described as follows.

$$G(t) = G_0 \cdot e^{-\alpha t/T} \quad (12)$$

where G_0 is the universal gravitational constant of the universe at the initial time t_0 , generally it is set as 100. α is 20, T is the maximum number of iterations and $R_{ij}(t)$ represented the Euclidean distance between object i and object j .

$$R_{ij} = ||X_i(t), X_j(t)|| \quad (13)$$

In GSA, the sum $F_i^k(t)$ of the forces acting on the X_i in the K -th dimension is equal to the sum of all the forces acting on this object:

$$F_i^k(t) = \sum_{j=1, j \neq i} rank_j F_{ij}^k(t) \quad (14)$$

where $rank_j$ is the random number in the range $[0, 1]$, $F_{ij}^k(t)$ is the gravity of the j -th object acting on the i -th object in the k -th dimension space. According to Newton's Second Law, the acceleration of the i -th particle in the k -th dimension at the moment t is defined as follows:

$$a_i^k(t) = \frac{F_i^k(t)}{M(t)} \quad (15)$$

2.2.4. Change the Positions

In each iteration, the object position can be changed by calculating the acceleration, which is calculated by the following equations.

$$v_i^k(t+1) = rank_i \times v_i^k(t) + a_i^k(t) \quad (16)$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \quad (17)$$

2.3. Algorithm Flowchart

The detailed flowchart of the algorithm is shown in Figure 2, and the optimization procedure is described as follows.

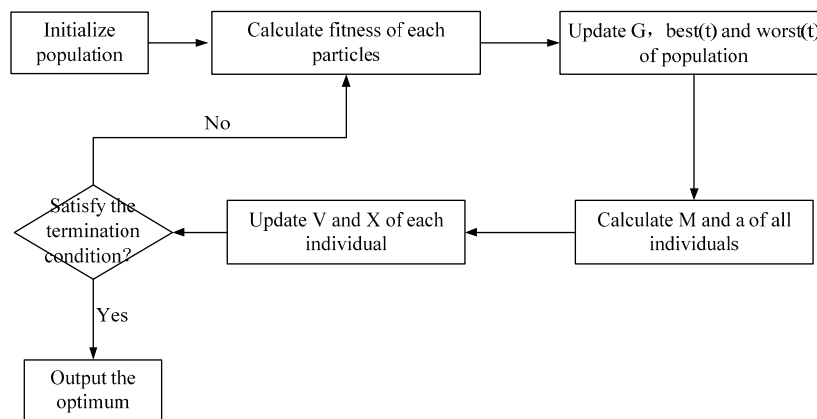


Figure 2. Flowchart of gravitational search algorithm (GSA).

Step 1: Initialize the positions and accelerations of all particles, the number of iterations and the parameters of the GSA;

Step 2: According to the Equation (12), calculate the fitness value of each particle and update the gravity constant;

Step 3: According to the Equations (5)–(7), calculate the quality of the particles based on the obtained fitness values and the acceleration of each particle according to the Equations (8) and (15);

Step 4: Calculate the velocity of each particle and update the position of the particle according to the Equation (17);

Step 5: If the termination condition is not satisfied, turn to Step 2, otherwise output the optimal solution.

2.4. Analysis of Gravitational Search Algorithm

In GSA, the update of the particle positions is aroused by the acceleration caused by the gravitational force of the particles. The value of gravity determines the size of the particle's acceleration, so the acceleration can be regarded as the search step of the particle position update. Its size determines the convergence rate of the gravitational search algorithm. In the acceleration calculation, the number of particles N and gravity play an important role, and the mass and gravity of the particles are determined the variety of G . The particle with different mass and gravity has different convergence rate. So it is very important to choose the number of particles and the gravity of the universal gravitation search algorithm.

The parameter initialization for all swarm intelligence optimization algorithms has important influence on the performance of the algorithms and the optimization ability. GSA has two main steps: one is to calculate the attraction of other particles for their selves and the corresponding acceleration calculated through gravitational, and another is to update the position of the particles according to the calculated acceleration. As shown in Formula (11)–(17), the convergence rate of GSA is determined by the value of Gravitational constant G_0 to determine the size of the particle acceleration, and

parameters α to determine the pace of change. In this section, the effect and influence of parameters G_0 and α will be analyzed in detail.

3. Simulation Experiments and Results Analysis

3.1. Test Functions

Six typical functions shown in Table 1 are selected and carried out simulation experiments for the optimal solution under the condition of GSA changeable parameters. There is only one extreme point for $f_1 \sim f_2$, which is mainly used to investigate the convergence of the algorithm and to test the accuracy of the algorithm. There are many extreme points for $f_3 \sim f_6$, the difference are in that $f_5 \sim f_6$ has lower dimension 2 and 4 respectively, while $f_1 \sim f_4$ are high dimension with 30.

Table 1. Typical test functions.

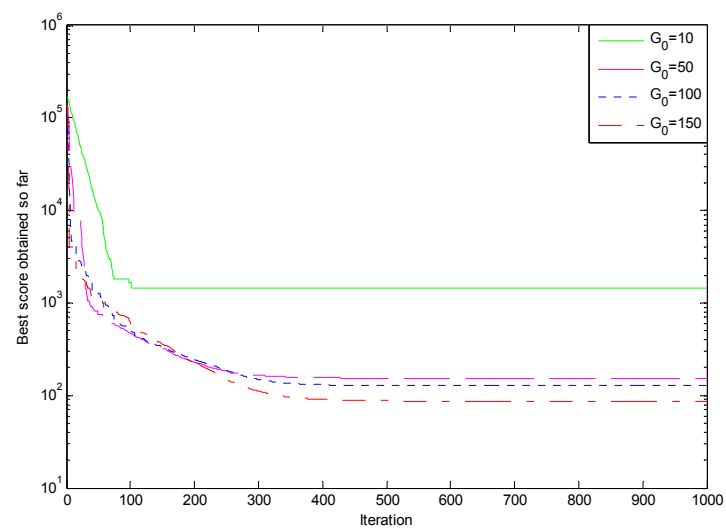
Function	Name	Expression	Range	Dim
f_1	Griewank	$\sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-32, 32]^n$	30
f_2	Quartic	$\sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]^n$	30
f_3	Rastrigin	$\sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	30
f_4	Schwefel	$418.9829n - \sum_{i=1}^n (x_i \sin \sqrt{ x_i })$	$[-600, 600]^n$	30
f_5	Shekel's Foxholes	$(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	$[-65.53, 65.53]^2$	2
f_6	Kowalik's	$\sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$[-5, 5]^4$	4

3.2. Simulation Results and Corresponding Analysis

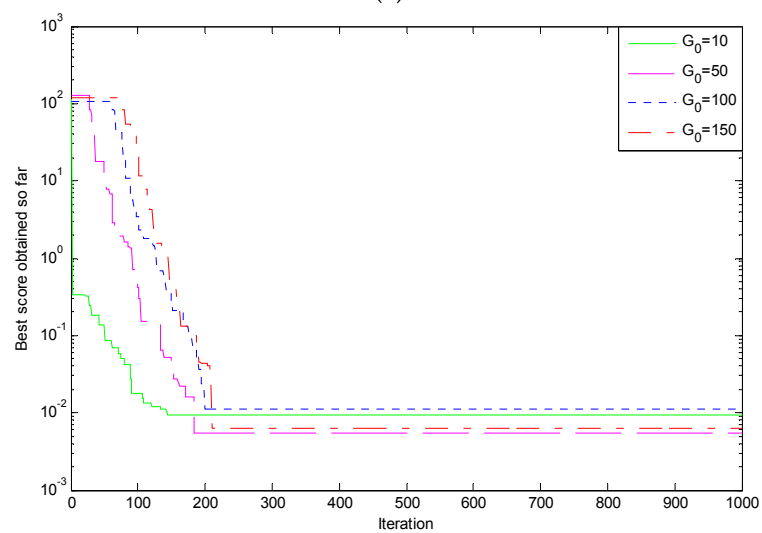
The GSA parameters are initialized as follows: max iterations max_it = 1000, objects N = 100. When the simulation experiments are carried out for the parameters G_0 , $\alpha = 20$. In order to reduce the influence of random disturbance, the independent operation 50 times is carried out for each test function. The optimum values and average values of GSA under different G_0 are shown in Table 2. The simulation curves for six test functions are shown in Figure 3a–f.

Table 2. The simulation result of different number of G_0 .

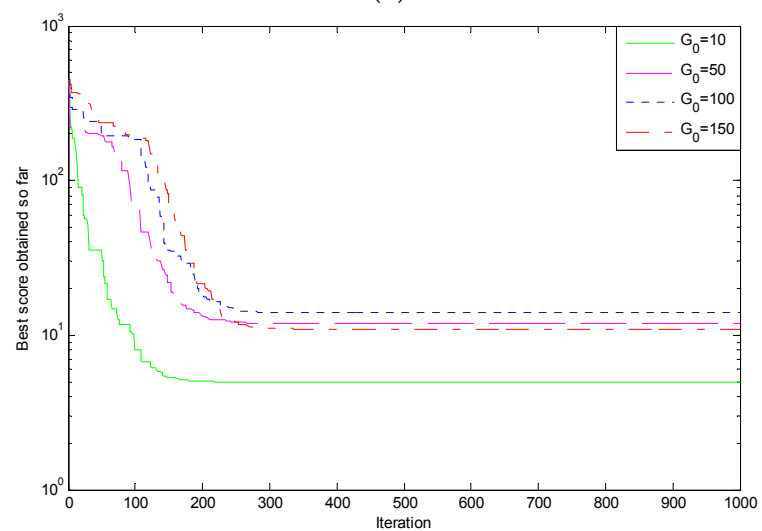
Function	Result	The Simulation Results of GSA under Different G_0				
		10	50	100	150	Minimum Time (s)
f_1	optimum	1.4465e+3	153.1636	127.7572	86.9076	9.1665
	average	216.2229	210.4132	640.4764	1.0409e+4	
f_2	optimum	0.0092	0.0055	0.0059	0.0062	7.6812
	average	69.0872	101.8753	102.9537	78.0762	
f_3	optimum	4.9748	11.9395	13.9294	10.9445	7.4739
	average	236.9825	392.0198	367.3913	365.4881	
f_4	optimum	315.2694	7.6050	2.4267	0.0123	7.7866
	average	31.1370	0	0	0	
f_5	optimum	1.0012	1.0909	1.0602	0.9980	7.3759
	average	2.0043	0.9980	1.0836	1.4385	
f_6	optimum	0.0013	0.0015	0.0017	0.0016	3.8364
	average	0.0072	0.0106	0.096	0.0049	



(a)

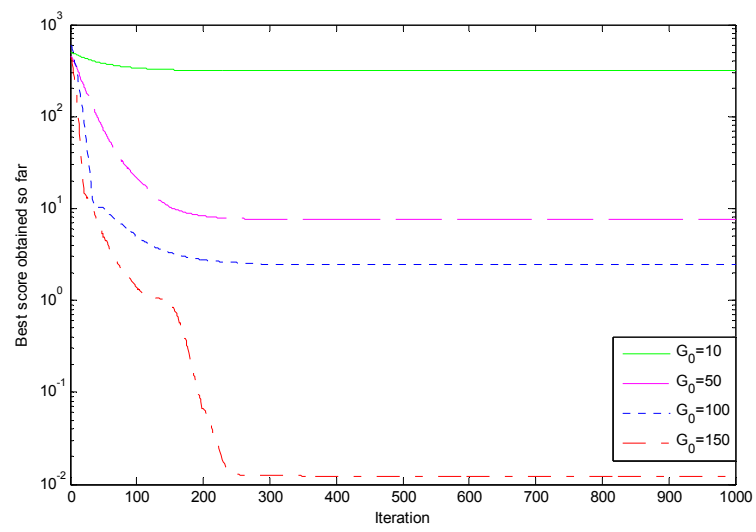


(b)

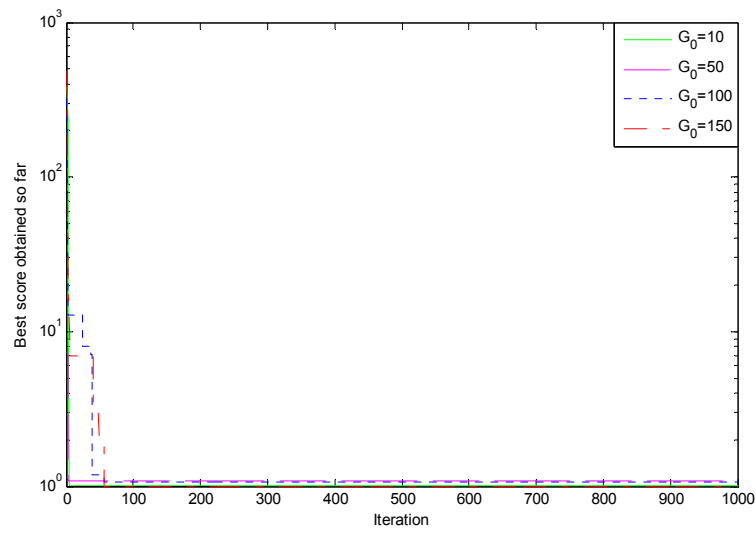


(c)

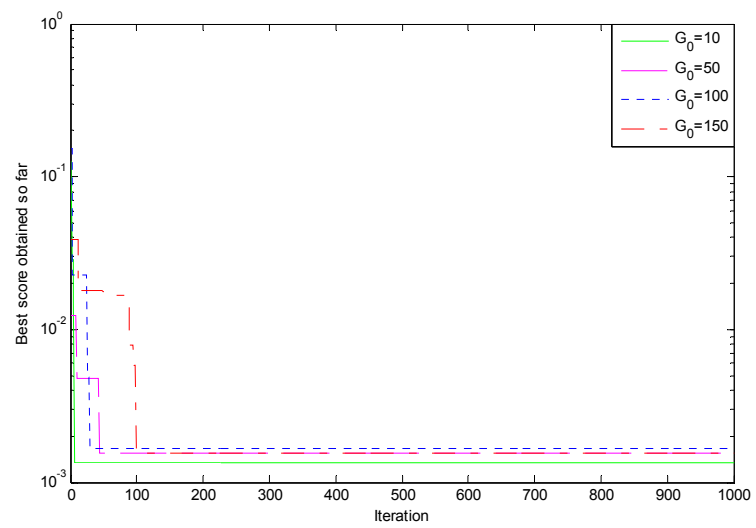
Figure 3. Cont.



(d)



(e)



(f)

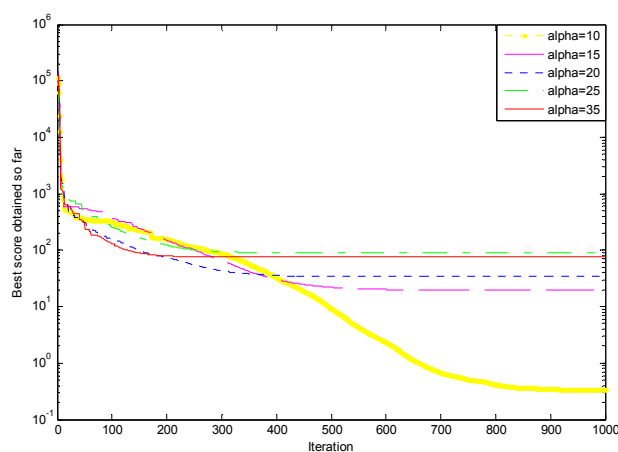
Figure 3. Simulation results of six typical test functions. (a) Griewank function; (b) Quartic function; (c) Rastrigin function; (d) Schwefel function; (e) Shekel's Foxholes function; (f) Kowalik's oxholes function.

Seen from the above simulation results, when G_0 is 150, whether the optimal value or average value, the functions (f_1 , f_2 , f_4) all have achieved the best results. When G_0 are 100 and 50, the optimization effect decreased successively. When G_0 is 10, the functions (f_3 , f_5 , f_6) have the best convergence performance. However, the most obvious performance of convergence curves appears in the function f_3 and the curves of optimization effect in function f_1 and f_4 is worst. The parameter selection of different simulation curve fluctuates greatly in f_4 , where the simulation curves of the difference maximum when G_0 is 10 compared with other values. By analyzing the convergence curves of each function, the convergence rate of the low dimensional function was higher than the high dimensional function. Seen from the overall trends, as the growth of the value G_0 , the precision of solution obtained in the related function optimization is not growth, but it is affected by the optimal solution of distribution from the solution space of a different function. Moreover, it is also related to the size of the solution space.

In view of the different values of the parameter α , the maximum or the minimum impact on the performance of function optimization are all varying. Considering the running time, the parameter G_0 is 100 and the other parameters remain unchanged. The simulation experiments and the corresponding analysis for α are shown in Table 3. The simulation curves of six test functions are shown in Figure 4a–f.

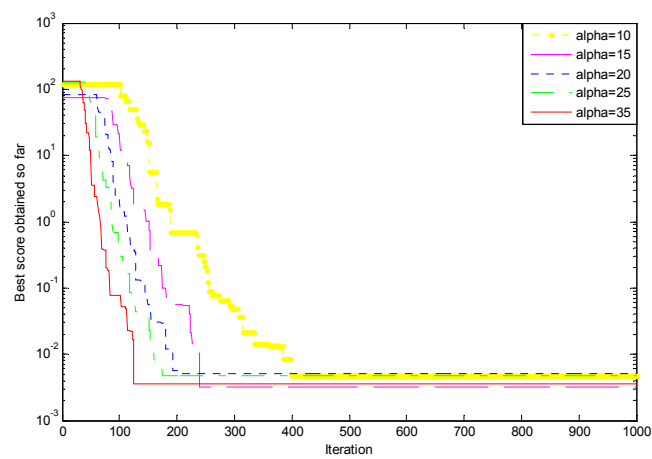
Table 3. The simulation result of different α .

Function	The Simulation Results of GSA under Different α					Minimum Time (s)
	10	15	20	25	35	
f_1	0.3386	19.7209	34.1272	89.9338	75.9954	31.9325
f_2	0.0047	0.0033	0.0050	0.0048	0.0036	27.9789
f_3	6.9647	4.9748	6.9647	5.9698	7.9597	27.9227
f_4	3.2581e−11	1.5543e−15	1.3438	2.5285	5.3956	28.4755
f_5	0.9980	0.9980	1.0064	3.9684	2.0038	19.3953
f_6	0.0011	0.0019	0.0022	0.0018	0.0024	13.9513

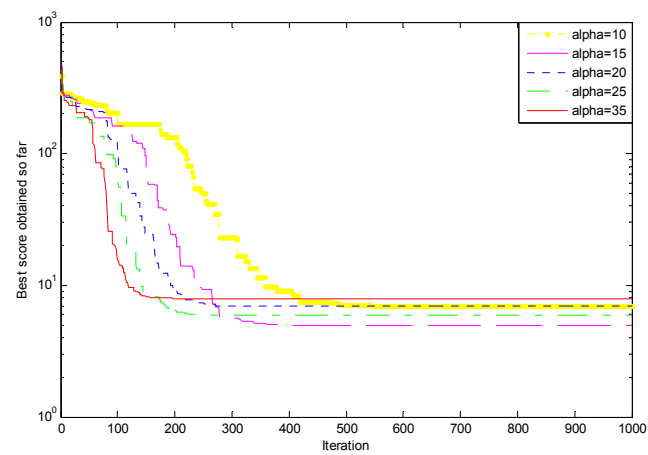


(a)

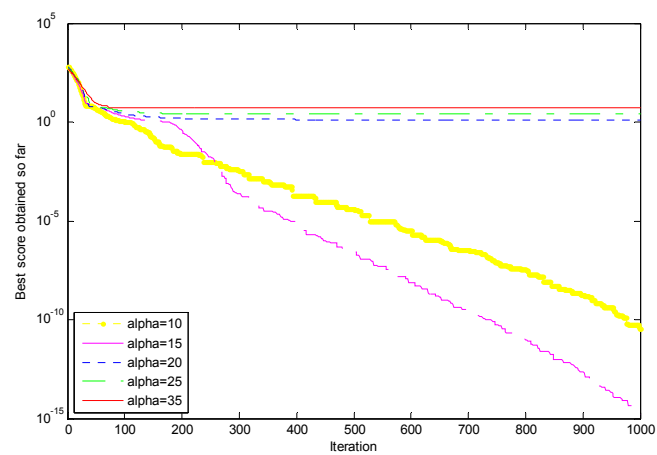
Figure 4. Cont.



(b)



(c)



(d)

Figure 4. Cont.

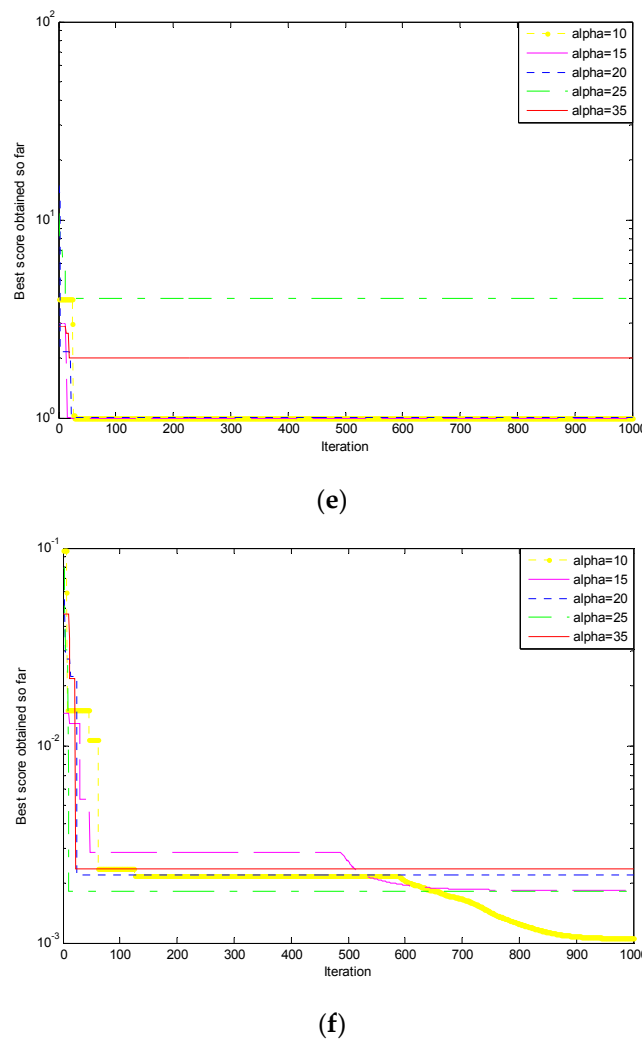


Figure 4. Simulation results of six typical test functions. (a) Griewank function f_1 ; (b) Quartic function f_2 ; (c) Rastrigin function f_3 ; (d) Schwefel function f_4 ; (e) Shekel's Foxholes function f_5 ; (f) Kowalik's Foxholes function f_6 .

Seen from the above simulation results, there are most times to obtain the optimal solution when the α value is 15, followed by 10. In addition to function f_1 , f_4 has a greater difference among the optimal solutions. The optimal solutions of other functions are close. The running time of low dimensional functions is less than the high dimensional functions. Comparing the shortest running times, the function f_6 is only a half of function f_1 , which shows that low dimensional function has better convergence performance. Seen from the points in convergence curves, when α is 10, the most obvious optimization performances of f_1 , f_4 , f_6 have converged to the optimal value. As the α increases, the convergence effect showed a decreasing trend and the convergence curves of the other three functions appear to show gradient optimization conditions. When α is 35, the functions f_2 , f_3 have the fastest convergence speed and reach the local optimal value early, but not the global optimal value. Overall, for different functions, the smaller α , the better the convergence performance. Compared to the slow convergence velocity, the functions with low dimension converge at a steeper point on the curve.

4. Conclusions

Based on the basic principle of the gravitational search algorithm (GSA), the algorithm flowchart is described in detail. The optimization performance is verified by simulation experiments on six test

functions. G_0 as a role of step in the particle position growth, through the simulation analysis of functions, when G_0 is 100, the algorithm is relatively inclined to be in a more stable state, which makes the optimal results more stable. The parameter α played a key role in control algorithm convergence rate. When the parameter value is small, the convergence speed of the algorithm is relatively slow. Under the same number of iterations, the conditions need for an optimal solution are worse; thus, a higher number of iterations is needed to obtain the optimal solution. However, the high value will easily cause the algorithm convergence speed to be too fast and become caught into the local solution, which will reduce the accuracy of the solution. As a result, the value 15 is appropriate. The simulation results show that the convergence speed of the algorithm is relatively sensitive to the setting of the algorithm parameters, and the GSA parameters can be used to improve the algorithm's convergence velocity and improve the accuracy of the solutions.

Acknowledgments: This work is partially supported by the National Key Technologies R & D Program of China (Grant No. 2014BAF05B01), the Project by National Natural Science Foundation of China (Grant No. 21576127), the Program for Liaoning Excellent Talents in University (Grant No. LR2014008), the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 2014020177), and the Program for Research Special Foundation of University of Science and Technology of Liaoning (Grant No. 2015TD04).

Author Contributions: Jie-Sheng Wang participated in the concept, design, interpretation and commented on the manuscript. Jiang-Di Song participated in the data collection, analysis, algorithm simulation, draft writing and critical revision of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ren, Y.; Wu, Y. An efficient algorithm for high-dimensional function optimization. *Soft Comput.* **2013**, *17*, 995–1004. [[CrossRef](#)]
2. Yuan, Z.; de Oca, M.A.M.; Birattari, M.; Stützle, T. Continuous optimization algorithms for tuning real and integer parameters of swarm intelligence algorithms. *Swarm Intell.* **2012**, *6*, 49–75. [[CrossRef](#)]
3. Mohan, B.C.; Baskaran, R. A survey: Ant Colony Optimization based recent research and implementation on several engineering domain. *Exp. Syst. Appl.* **2012**, *39*, 4618–4627. [[CrossRef](#)]
4. Vallada, E.; Ruiz, R. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *Eur. J. Oper. Res.* **2011**, *211*, 612–622. [[CrossRef](#)]
5. Kennedy, J. *Particle Swarm Optimization*; Springer: Berlin, Germany, 2010; pp. 760–766.
6. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [[CrossRef](#)]
7. Zandi, Z.; Afjei, E.; Sedighzadeh, M. Reactive Power Dispatch Using Big Bang-Big Crunch Optimization Algorithm for Voltage Stability Enhancement. In Proceedings of the IEEE International Conference on Power and Energy (PECon), Kota Kinabalu, Malaysia, 2–5 December 2012; pp. 239–244.
8. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Inf. Sci.* **2013**, *222*, 175–184. [[CrossRef](#)]
9. Formato, R.A. Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Prog. Electromagn. Res.* **2007**, *77*, 425–491. [[CrossRef](#)]
10. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [[CrossRef](#)]
11. Cuevas, E.; Oliva, D.; Zaldivar, D.; Pérez-Cisneros, M.; Sossa, H. Circle detection using electro-magnetism optimization. *Inf. Sci.* **2012**, *182*, 40–55. [[CrossRef](#)]
12. Shah-Hosseini, H. Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation. *Int. J. Comput. Sci. Eng.* **2011**, *6*, 132–140. [[CrossRef](#)]
13. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
14. Shah-Hosseini, H. Problem Solving by Intelligent Water Drops. In Proceedings of IEEE Congress on the Evolutionary Computation, Singapore, 25–28 September 2007; pp. 3226–3231.
15. Rabanal, P.; Rodríguez, I.; Rubio, F. *Using River Formation Dynamics to Design Heuristic Algorithms*; Springer Berlin Heidelberg: Berlin, Germany, 2007; pp. 163–177.

16. Vicsek, T.; Czirók, A.; Ben-Jacob, E.; Cohen, I.; Shochet, O. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **1995**, *6*, 1226–1229. [[CrossRef](#)] [[PubMed](#)]
17. Tamura, K.; Yasuda, K. Primary study of spiral dynamics inspired optimization. *IEEE Trans. Electr. Electron. Engin.* **2011**, *6*, S98–S100. [[CrossRef](#)]
18. Eskandar, H.; Sadollah, A.; Bahreininejad, A.; Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **2012**, *10*, 151–166. [[CrossRef](#)]
19. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci. Int. J.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
20. Hossein, H.; Ehsanolah, A.; Assareh, E. Optimization of hybrid laminated composites using the multi-objective gravitational search algorithm (MOGSA). *Eng. Optim.* **2014**, *9*, 1169–1182.
21. Beatriz, G.; Fevrier, V.; Melin, P.; Prado-Arechiga, G. Fuzzy logic in the gravitational search algorithm for the optimization of modular neural networks in pattern recognition. *Expert Syst. Appl.* **2015**, *8*, 5839–5847.
22. Dowlatshahi, M.B.; Nezamabadi-Pour, H. GGSA: A Grouping Gravitational Search Algorithm for data clustering. *Eng. Appl. Artif. Intell.* **2014**, *11*, 114–121. [[CrossRef](#)]
23. Su, Z.; Wang, H. A novel robust hybrid gravitational search algorithm for reusable launch vehicle approach and landing trajectory optimization. *Neurocomputing* **2015**, *8*, 116–127. [[CrossRef](#)]
24. Duman, S.; Yorukeren, N.; Altas, I.H. A novel modified hybrid PSOGSA based on fuzzy logic for non-convex economic dispatch problem with valve-point effect. *Int. J. Electr. Power Energy Syst.* **2015**, *6*, 121–135. [[CrossRef](#)]
25. Nezamabadi-pour, H. A quantum-inspired gravitational search algorithm for binary encoded optimization problems. *Eng. Appl. Artif. Intell.* **2015**, *4*, 62–75. [[CrossRef](#)]
26. Mirjalili, S.; Wang, G.G.; Coelho, L.S. Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Comput. Appl.* **2014**, *25*, 1423–1435. [[CrossRef](#)]
27. Yuan, X.; Ji, B.; Zhang, S.; Tian, S.; Hou, Y. A new approach for unit commitment problem via binary gravitational search algorithm. *Appl. Soft Comput.* **2014**, *9*, 249–260. [[CrossRef](#)]
28. Ganesan, T.; Elamvazuthi, I.; Shaari, K.K.; Vasant, P. Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production. *Appl. Energy* **2013**, *103*, 368–374. [[CrossRef](#)]
29. Zhang, W.; Niu, P.; Li, G.; Li, P. Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm. *Knowl. Based Syst.* **2013**, *39*, 34–44. [[CrossRef](#)]
30. Pal, K.; Saha, C.; Das, S.; Coello, C.A.C. Dynamic Constrained Optimization with Offspring Repair Based Gravitational Search Algorithm. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Cancun, Mexico, 20–23 June 2013; pp. 2414–2421.
31. David, R.C.; Precup, R.E.; Petriu, E.M.; Rădac, M.B.; Preitl, S. Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity. *Inf. Sci.* **2013**, *247*, 154–173. [[CrossRef](#)]
32. Yong, L.; Liang, M. Improved Gravitational Search Algorithm for Grey Nonlinear Constrained Programming Problem. *Math. Pract. Theory* **2013**, *43*, 99–103.
33. Shaw, B.; Mukherjee, V.; Ghoshal, S.P. Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm. *Int. J. Electr. Power Energy Syst.* **2014**, *55*, 29–40. [[CrossRef](#)]
34. Liu, Y.; Ma, L. Gravitational Search Algorithm for Minimum Ratio Traveling Salesman Problem. *J. Chin. Comput. Syst.* **2013**, *34*, 847–849.
35. Li, C.; Li, H.; Kou, P. Piecewise function based gravitational search algorithm and its application on parameter identification of AVR system. *Neurocomputing* **2014**, *124*, 139–148. [[CrossRef](#)]
36. Kumar, J.V.; Kumar, D.M.V.; Edukondalu, K. Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market. *Appl. Soft Comput.* **2013**, *13*, 2445–2455. [[CrossRef](#)]

