A Hybrid Algorithm Based on Gravitational Search and Particle Swarm Optimization Algorithm to Solve Function Optimization Problems

Jie-Sheng Wang, and Jiang-Di Song

Abstract—**Gravitational search algorithm (GSA) is a swarm intelligence heuristic optimization algorithm based on the law of gravitation. Aiming at the disadvantage of poor local search ability and slow convergence speed in standard GSA, four improved GSA-PSO hybrid algorithm are proposed by introducing a small constant updating strategy in order to enhance the update ability of velocity, acceleration factor and the optimal individual location, where PSO strategy was used to optimize the position and velocity of the GSA. Through simulation experiments on typical test functions to verify its performance, the simulation results show that the optimal setup of GSA parameters can improve the convergence rate of the algorithm and improve the accuracy of the solution.**

Index Terms—**gravitational search algorithm, particle swarm optimization algorithm, function optimization**

I. Introduction

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 function. There has been a considerable attention paid for 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 the 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], Bat Algorithm (BA) [5], Artificial Bee Colony (ABC) algorithm [6], etc.

But, not all metaheuristic 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 metaheuristic 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], which 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. It has been successfully applied in many global optimization problems, such as, multi-objective optimization of synthesis gas production [20]，forecasting of turbine heat rate [21]，dynamic constrained optimization with offspring repair [22]，fuzzy control system [23]，grey nonlinear constrained programming problem [24]，reactive power dispatch of power systems [25]，minimum ratio traveling salesman problem [26]，parameter identification of AVR system [27]，strategic bidding [28], etc.

In this paper, four kinds of improved GSA-PSO hybrid algorithm were proposed by introducing a small constant updating mechanism, which adopts PSO strategy to optimize the velocity and position in the running process of the GSA. The simulation analysis results show that the improved hybrid algorithm greatly improves the function optimization convergence speed and optimization accuracy.

Manuscript received June 17, 2016; revised October 5, 2016. This work was supported by 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), the Program for Research Special Foundation of University of Science and Technology of Liaoning (Grant No. 2015TD04) and the Opening Project of National Financial Security and System Equipment Engineering Research Center (Grant No. USTLKFGJ201502 and USTLKEC201401).

Jie-Sheng Wang is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China; National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning. (phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Jiang-Di Song is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: sjd2011@163.com).

(Advance online publication: 22 February 2017)

II. Gravitational Search Algorithm

A. Physics foundation of GSA

The law of universal gravitation is one of the four basic 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:

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

where, F is the gravitational force between two objects, G is the gravitational constant, M_1 and M_2 are the masses of the object 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 $6.67\times10^{-11} N\cdot m^2/kg^2$.

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

$$a = \frac{F}{M} \tag{2}$$

According to the Eq. (1) and (2), all of the particles in the world are affected by gravity. The more close the distance between two particles, the greater the gravitational. 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.

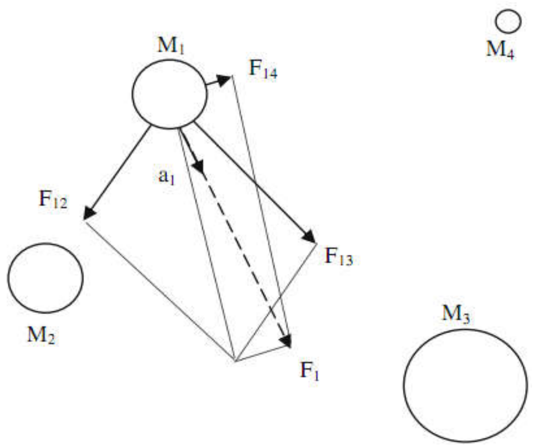


Fig. 1. Gravitational phenomena

B. Basic principles of gravitation search algorithm

1) Initialize the locations

Because of no need to consider the environmental impact, the position of a particle is initialized as X_i. Firstly, randomly generate the positions $x_i^1, x_i^2, ..., x_i^k, ... x_i^d$ of N objects, and then the positions of N objects are brought into the function, where the position of the i th object is defined as follows.

$$X_i = (x_i^1, x_i^2, ... x_i^k, ..., x_i^d) \tag{3}$$

2) Calculate the inertia mass

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.

$$M_{ai} = M_{pi} = M_{ii} = M_i \tag{4}$$

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

$$M_i(t) = \frac{m_i(t)}{\sum^n m_j(t)} \tag{6}$$

where $i = 1, 2, \cdots, n$, $fit(i)$ 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:

$$best(t) = \max_{j=\{1,2,...,n\}} fit_j(t) \tag{7}$$

$$worst(t) = \min_{j=\{1,2,...,n\}} fit_j(t) \tag{8}$$

For solving the minimum value problem:

$$best(t) = \min_{j=\{1,2,...,n\}} fit_j(t) \tag{9}$$

$$worst(t) = \min_{j=\{1,2,...,n\}} fit_j(t) \tag{10}$$

3) Calculate gravitational force

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

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

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.

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

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 $R_{ij}(t)$ represents the Euclidean distance between object i and object j.

$$R_{\mathrm{ij}} = \left\| X_i(t), X_j(t) \right\| \tag{13}$$

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:

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

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 the Newton's Second Law, the acceleration of the i th particle in the k th dimension at the moment t is defined as follows:

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

4) Change the positions

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

$$v_i^k(t+1) = rank_i \times v_i^k(t) + a_i^k(t) \tag{16}$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \tag{17}$$

C. Algorithm flowchart

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

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 Eq. (12), calculate the fitness value of each particle and update the gravity constant;

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

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

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

III. GSA-PSO Hybrid Optimization Algorithm

A. Particle Swarm Optimization Algorithm

PSO algorithm adopts a large number of particles to search the optimum in the space solution, where the velocity of the particle depends on the direction and distance of its flight, and they have their own fitness value. The dynamic adjustment of each particle to find the optimal solution relies on the individual extreme and the global extreme, until the termination is met [29].

The location of the i th particle is represented as $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$. At each generation, each particle is updated by following the two 'best' values. The best previous position of the i th particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, which is also called $pBest$. The index of the best $pBest$ among all the particles is represented by the symbol g. The location $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$ is also called $gBest$. The velocity for the i th particle is represented as $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$. The PSO concept consists of, at each generation, updating the velocity and location of each particle toward its $pBest$ and $gBest$ locations according to the Eq. (16) and (17), respectively:

$$\begin{aligned} V_{id}^{n+1} = \omega\ V_i^t + c1\ \times rand \times (pbest_i - x_i^t) + \\ c2\ \times rand \times (gbest_i - x_i^t) \end{aligned} \tag{16}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{17}$$

where w is inertia weight, c_1 and c_2 are learning constants, and $rand()$ is a random function in the range [0, 1]. For Eq. (16), the first part represents the inertia of the previous velocity; the second part is the 'cognition' part, which represents the private thinking by itself; the third part is the 'social' part, which represents the co-operation among the particles. V_{id} is clamped to a maximum velocity $V_{\max}$.

B. GSA-PSO Algorithm

In recent years, Seyedali M. proposed BPSOGSA hybrid algorithm with binary parameters of the PSOGSA algorithm [29] in order to solve the binary parameter optimization problem. Four hybrid algorithms (GSAPSO-1, GSAPSO-2, GSAPSO-3 and GSAPSO-4) based on the standard GSA-PSO algorithm in the literature [29] are proposed, which adopts the small constant ε updating strategy in three parts on the on the material motion equation respectively so as to the memory function when updating the velocity, acceleration, and the optimal positions. The detailed analysis are described as follows.

The strong exploration ability of GSA and the PSO algorithm are combined to obtain the better optimization ability. The improved particle velocity updating equations are described as follows.

$$V_i(t+1) = r \times\ V_i^t + c_1^{'}\ \times r \times ac_i(t) + c_2^{'}\ \times r \times (gbest_i - x_i(t)) \tag{18}$$

$$V_i(t+1) = (r+\varepsilon) \times\ V_i^t + c_1^{'}\ \times r \times ac_i(t) + c_2^{'}\ \times r \times (gbest_i - x_i(t)) \tag{19}$$

$$V_i(t+1) = r \times\ V_i^t + c_1^{'}\ \times (r+\varepsilon) \times ac_i(t) + c_2^{'}\ \times r \times (gbest_i - x_i(t)) \tag{20}$$

$$V_i(t+1) = r \times\ V_i^t + c_1^{'}\ \times r \times ac_i(t) + c_2^{'}\ \times (r+\varepsilon) \times (gbest_i - x_i(t)) \tag{21}$$

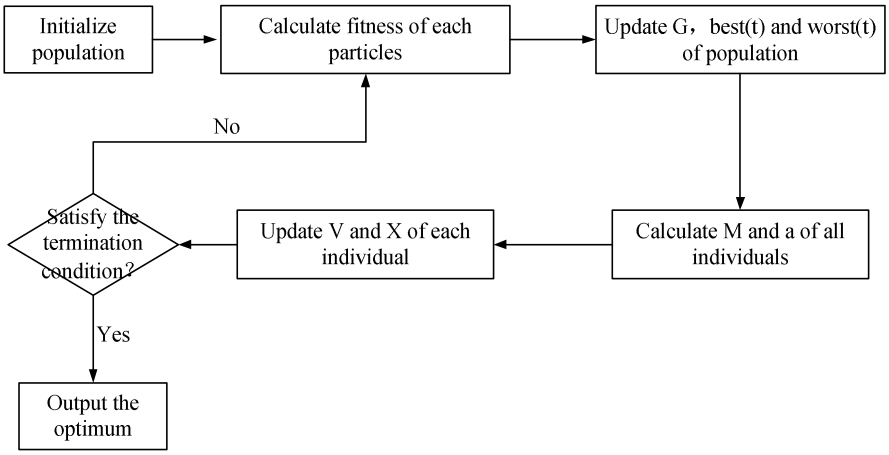


Fig. 2. Flowchart of GSA

where V_{it} is the velocity of particle i at the generation t, c'_j is the acceleration coefficient, r is a random located in the scope [0,1], ε is a small constant, $ac_i(t)$ is the acceleration of particle i at the generation t, and $gbest$ is the optimal solution so far.

The position of the particles is updated according to the following equation.

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{22}$$

It is worth noting that the quality and inertia mass equations of particle i at the generation t are improved in the literature (29), which is described as follows, respectively.

$$m_i(t) = \frac{fit_i(t) - 0.99 \times worst(t)}{best(t) - worst(t)} \tag{23}$$

$$M_i(t) = \frac{5 \times m_i(t)}{\sum_{j=1}^{n} m_j(t)} \tag{24}$$

C. Algorithm Procedure

The algorithm procedure of the improved GSA-PSO algorithm shown in Figure 3 is described as follows.

Step 1: Initialize the algorithm parameters, including the total number of particles N, the number of iterations t, the gravitational constant G_0 and the decreasing coefficient α.

Step 2: Randomly generate populations. The position vector of the particle is set as $X = (x_1, x_2, \ldots, x_n)$, the velocity is initialized as $v_i = (v_{i1}, v_{i2}, \ldots, v_{is})^T$, the global optimal value is $gbest$ and the individual optimal value is $pbest$.

Step 3: Calculate the fitness value of each individual $fit(t)$, find the best fitness value $best(t)$ and the worst fitness value $worst(t)$, and record the best position $gbest$.

Step 4: Calculate the quality of each individual according to Eq. (5)-(8); then calculate the gravitational constant G and the Euclidean distance between two particles so as to calculate the individual's gravity F and acceleration a. At this time, by adopting the global search ability of PSO algorithm, four hybrid algorithms are used to update the position and velocity of the individual by using Eq. (18)-(22); then calculate the fitness value $fit(t)$ and the optimal value $gbest'$ of each particle in the improved algorithms. $gbest'$ is compared with the $gbest$ generated in Step 3 to obtain the optimal value $gbest''$.

Step 5: Judge whether the termination function or the number of iterations is met. If it is still not satisfied, return to Step 3.

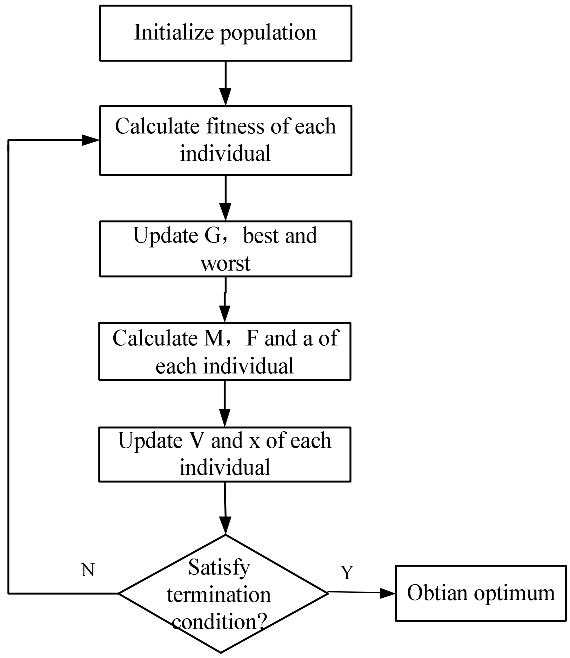


Fig. 3 The flowchart of improved GSAPSO hybrid algorithm

IV. Simulation Experiments and Results Analysis

The parameters of the hybrid algorithm are initialized as follows: object number N =30, the maximum number of iterations max_it=1000, the gravitational constant G_0 =100 at time t0, α α =23. Four test functions are adopted to carry out simulation and software MATLAB is used as the simulation platform. The independent operations are carried out 50 times. Four test functions are shown in Table 1, which are named as F1, F2, F3 and F4. The dimension of F1, F2 and F3 is 30, and the dimension of F4 is 2. On the other hand, F1 is a single function, the others are polymorphic functions.

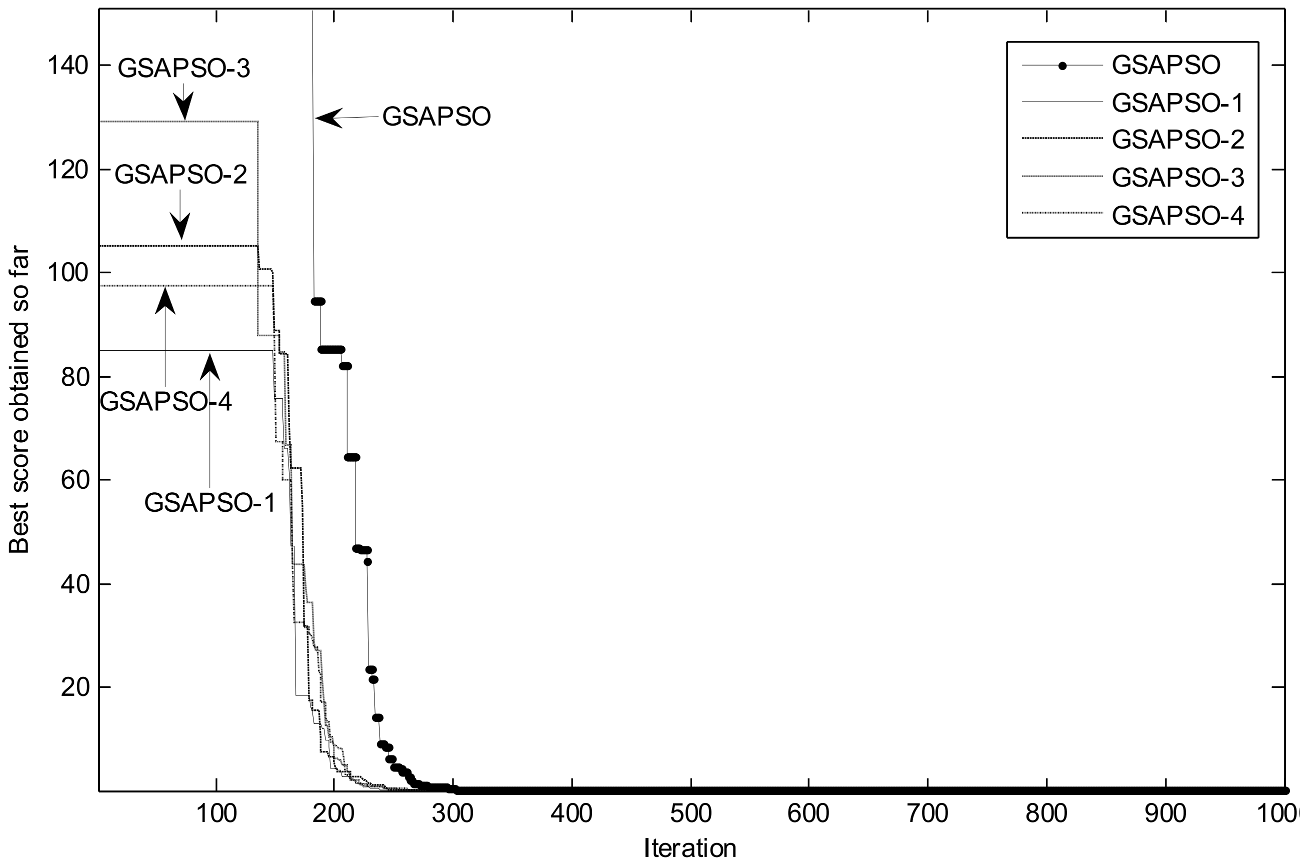
Figure 4 (a) - (b) are the compared optimization results of the above proposed algorithms. In order to make the simulation results more clearly, for the F3 and F4 with fast convergence speed, the 100 generation simulation results are retained within the optimization curves. The simulation statistical results are shown in Table 2.

It can be seen from the above simulation results that the optimization time of low dimensional function F4 is smaller than other high dimensional function. The fast convergence rate makes each algorithm gradually tend to the optimal value and reach the steady state in the iteration number of 15-60. In addition, the optimal values of the complex multi-state function F3 and F4 are equal in each hybrid algorithm. Comparison of the optimization results, four kinds of hybrid algorithm are better than the GSA-PSO algorithm proposed in the literature (29). For the unimodal function F1, the optimization results are compared as: GSAPSO-2 < GSAPSO-4 < GSAPSO-3 < GSAPSO-1. For the multi peak function F2, the optimization results are compared as: GSAPSO-3 < GSAPSO-2 < GSAPSO-1 < GSAPSO-4.

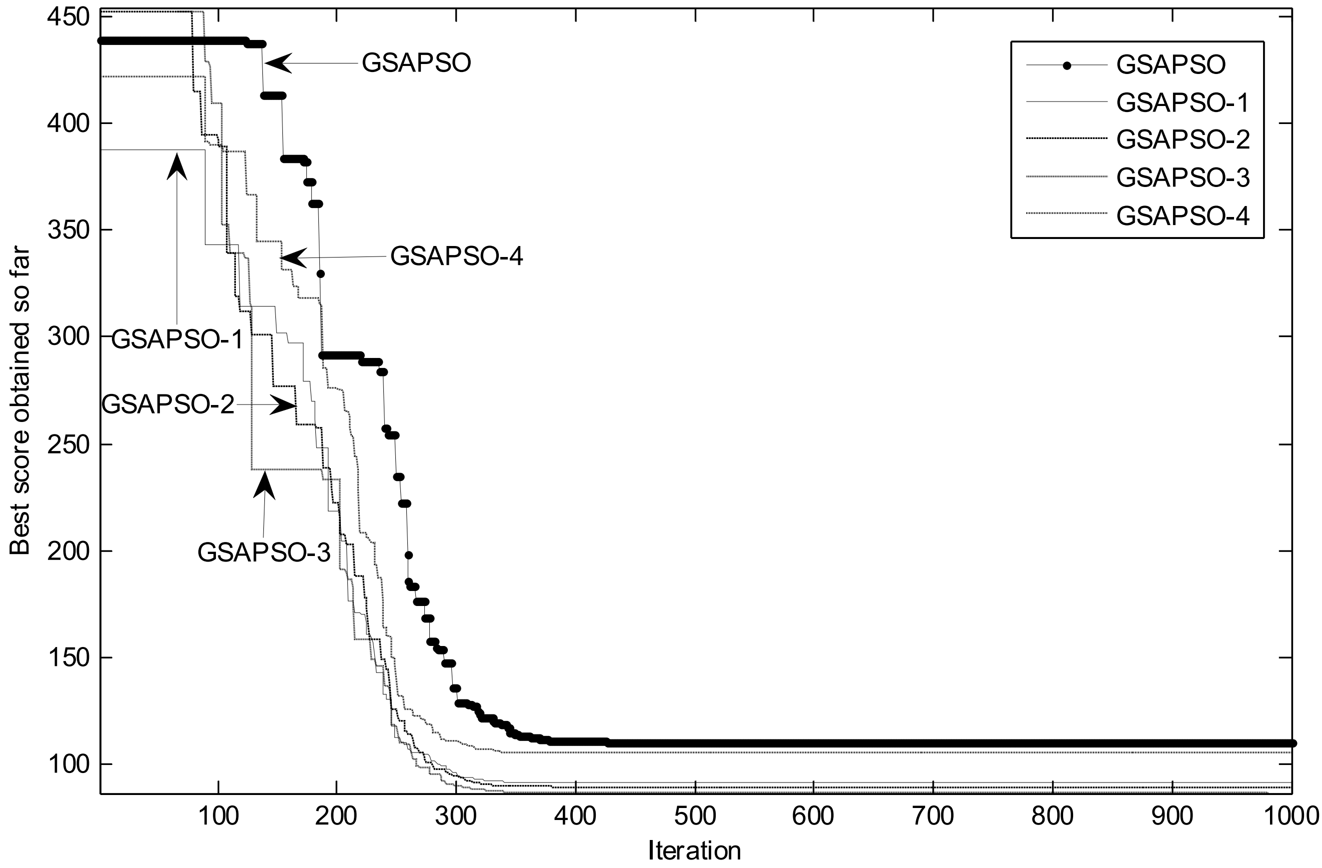
In conclusion, through the introduction of the small constant update strategy, the optimization performances of four proposed methods are improved significantly. For single state function, the ability to update the speed and position of the best individual in the hybrid algorithm is enhanced, which can effectively adjust the influence of "law of attraction" and "social information exchange" on the particles. For the multi-peak functions, the acceleration and velocity at the particle movement by "law of attraction" and "speed" function are adjusted. In short, the improved hybrid algorithms have an important influence on the law of attraction between substances.

TAB. 1 SIMULATION FUNCTIONS

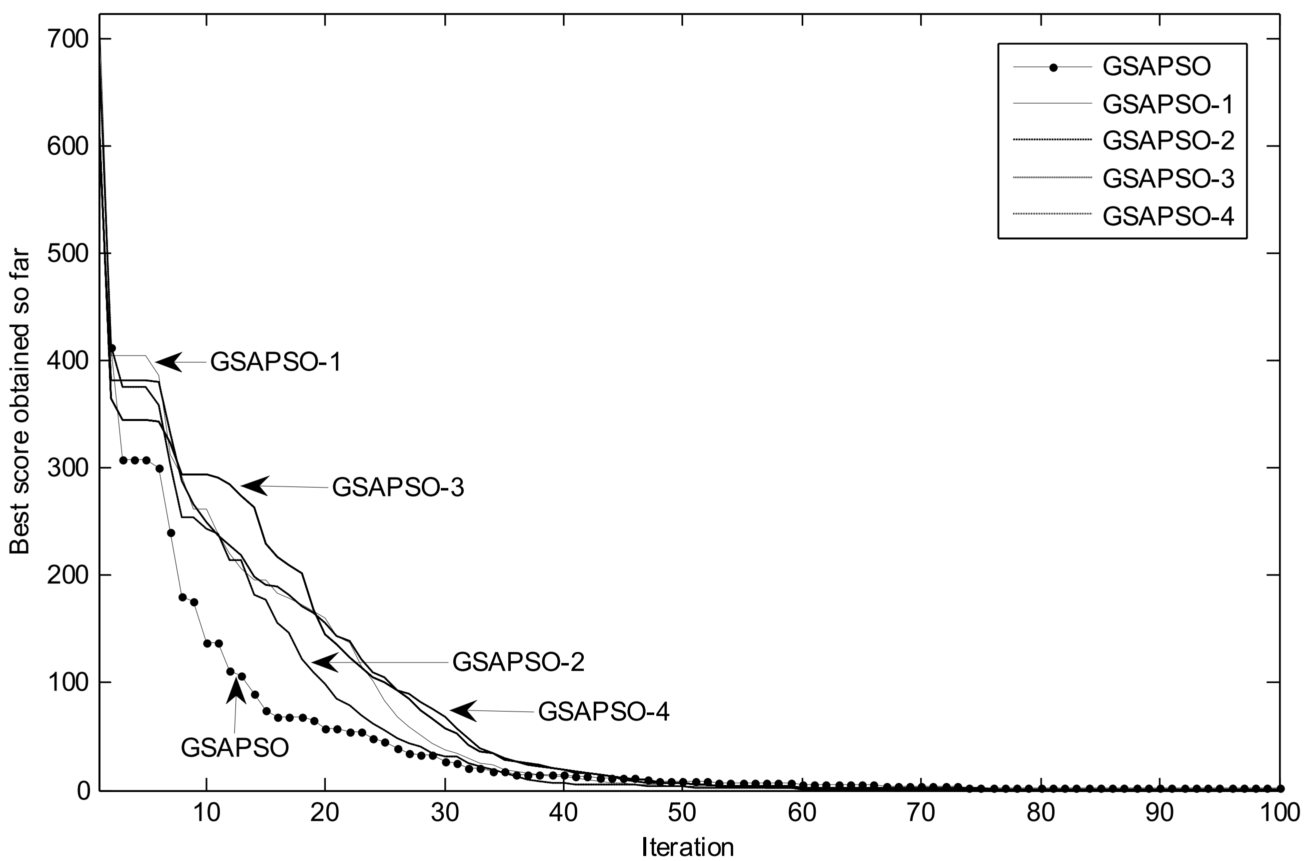
Function	Name	Expression equation	Scope	Dimension
F_1	Quartic	$\sum_{i=1}^{n} ix_i^4 + random[0,1)$	$[-1.28,1.28]^n$	30
F_2	Rastrigin	$\sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$[-5.12,5.12]^n$	30
F_3	Schwefel	$418.9829n - \sum_{i=1}^{n}\left(x_i \sin\sqrt{\lvert x_i\rvert}\right)$	$[-600,600]^n$	30
F_4	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



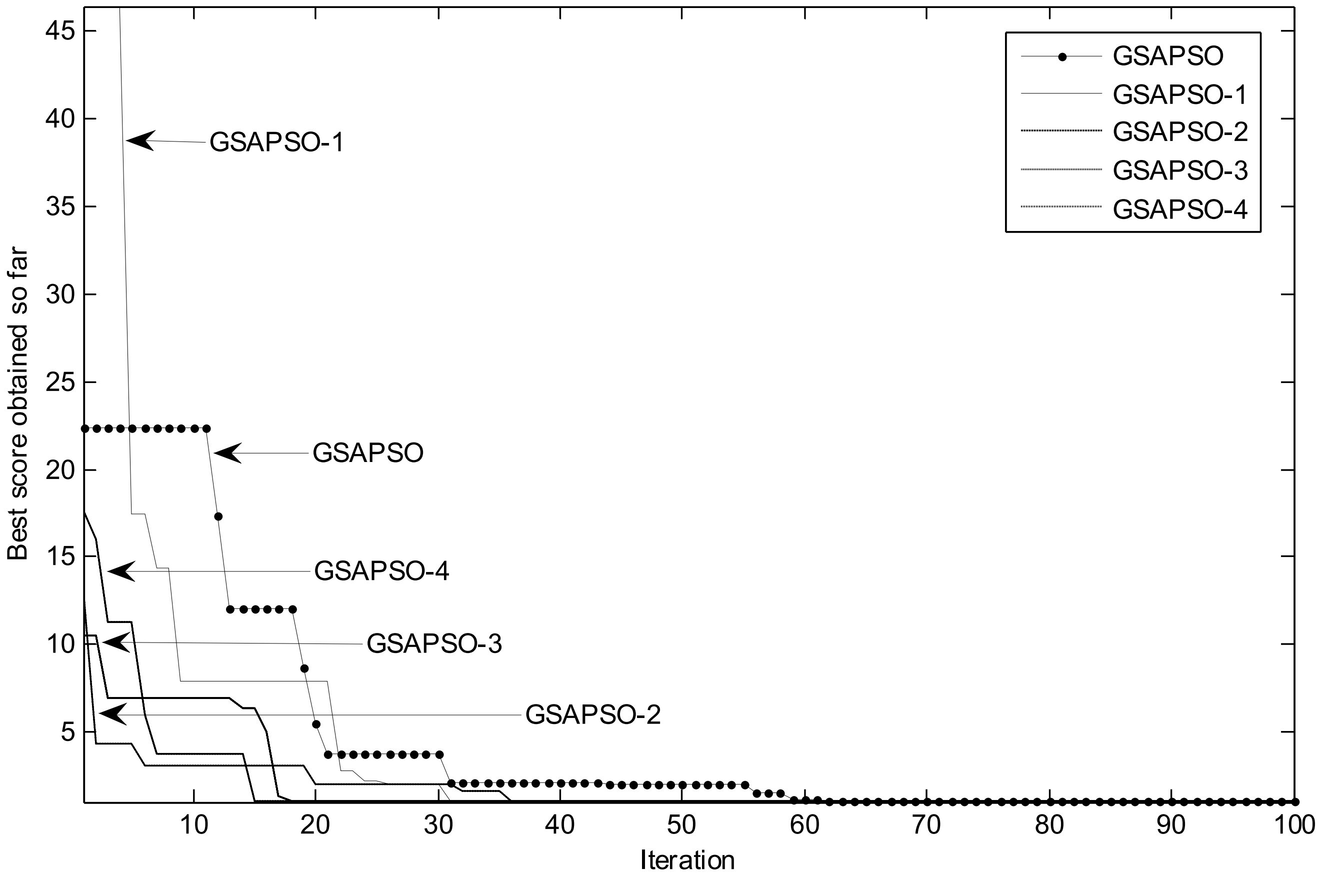
(a) Quartic



(b) Rastrigin



(c) Schwefel



(d) Shekel's Foxholes function

Fig. 4 Simulation curves of five algorithms for function optimization

TAB. 2 THE SIMULATION RESULTS OF GSAPSO HYBRID ALGORITHM

Function / Algorithm		GSAPSO	GSAPSO-1	GSAPSO-2	GSAPSO-3	GSAPSO-4
F_1	Optimum	0.0494	0.0455	0.0281	0.0409	0.0352
	Time (s)	4.718	4.912	4.788	4.720	5.054
F_2	Optimum	110.439	91.535	89.545	86.561	105.465
	Time (s)	4.137	4.649	4.649	4.852	4.657
F_3	Optimum	0	0	0	0	0
	Time (s)	4.189	4.334	4.226	4.358	4.224
F_4	Optimum	0.9980	0.9980	0.9980	0.9980	0.9980
	Time (s)	3.572	3.642	3.666	3.737	3.711

V. CONCLUSION

Gravitational search algorithm (GSA) is a swarm intelligence heuristic optimization algorithm based on the law of gravitation. In this paper, four improved GSA-PSO hybrid algorithm are proposed by introducing the optimization strategy of the motion equation based on the micro constant renewal material. The simulation results show that the improved algorithm has good performance.

REFERENCES

[1] Y. Ren, and Y. Wu, "An efficient algorithm for high-dimensional function optimization." *Soft Computing*, vol. 17, no. 6, pp. 995–1004, Jun. 2013.

[2] Z. Yuan, M. A. M. de Oca, and M. Birattari, "Continuous optimization algorithms for tuning real and integer parameters of swarm intelligence algorithms," *Swarm Intelligence*, vol. 6, no. 1, pp. 49–75, Mar. 2012.

[3] Y. Ghanou, and G. Bencheikh, "Architecture Optimization and Training for the Multilayer Perceptron using Ant System," *IAENG International Journal of Computer Science*, vol. 43, no.1, pp. 20-26, 2016.

[4] E. Vallada, and R. Ruiz, "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times," *European Journal of Operational Research*, vol. 211, no. 3, pp. 612–622, Jun. 2011.

[5] Anping Song, Mingbo Li, Xuehai Ding, Wei Cao, and Ke Pu, "Community Detection Using Discrete Bat Algorithm," *IAENG International Journal of Computer Science*, vol. 43, no.1, pp37-43, 2016.

[6] Chun-Feng Wang, and Yong-Hong Zhang, "An Improved Artificial Bee Colony Algorithm for Solving Optimization Problems," *IAENG International Journal of Computer Science*, vol. 43, no.3, pp336-343, 2016.

[7] Z. Zandi, E. Afjei, and M. Sedighizadeh, "Reactive power dispatch using big bang-big crunch optimization algorithm for voltage stability enhancement," *in Proc. IEEE International Conference on Power and Energy)*, Kota Kinabalu, Malaysia, 2012, pp. 239–244.

[8] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, no. 3, pp. 175–184, Feb. 2012.

[9] R. A. Formato, "Central force optimization: A new metaheuristic with applications in applied electromagnetics," *Progress In Electromagnetics Research*, vol. 77, no. 1, pp. 425–491, Jan. 2007.

[10] A. Kaveh, and S Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3-4, pp. 267–289, Sep. 2010.

[11] E. Cuevas, D. Oliva, D. Zaldivar, M. Pe´rez-Cisneros, and H. Sossa, "Circle detection using electro-magnetism optimization," *Information Sciences* , vol. 182, no. 1, pp. 40–55, Jan. 2012.

[12] Shah-Hosseini, and Hamed, "Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization," *International Journal of Computational Science and Engineering*, vol. 6, no. 1, pp. 132–140, Jul. 2011.

[13] W. G. Zong, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation Transactions of the Society for Modeling & Simulation International*, vol. 76, no. 2, pp. 60–68, Feb. 2001.

[14] H. Shah-Hosseini, "Problem solving by intelligent water drops. In Evolutionary Computation," *in Proc. 2007 IEEE Congress on Evolutionary Computation (CEC)*, Singapore, 2007, pp. 3226–3231.

[15] P. Rabanal, I. Rodríguez, and F. Rubio, "Using River Formation Dynamics to Design Heuristic Algorithms," *in Proc. 6th International Conference on Unconventional Computation*, Kingston, Canada, 2007, pp. 163–177.

[16] T. Vicsek, A. Cziro´k, E. Ben-Jacob, I. I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, pp. 1226–1229, Aug. 2006.

[17] K. Tamura, and K. Yasuda, "Primary study of spiral dynamics inspired optimization," *Ieej Transactions on Electrical & Electronic Engineering*, vol. 6, no. S1, pp. S98–S100, Nov. 2011.

[18] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. S110-111, no. 10, pp. 151–166, Nov. 2012.

[19] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[20] T. Ganesan, I. Elamvazuthi, K. Z. K. Shaari, and P. Vasant, "Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production," *Applied Energy*, vol. 3, no. 3, pp. 280–292, Mar. 2013.

[21] W. Zhang, P. Niu, G. Li, and P. Li, "Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm," *Knowledge-Based Systems*, vol. 39, no. 2, pp. 34–44, Feb. 2013.

[22] K. Pal, C. Saha, S. Das, and CAC Coello, "Dynamic Constrained Optimization with offspring repair based Gravitational Search Algorithm," *in Proc. 2013 IEEE Congress on Evolutionary Computation (CEC),* Cancun, Mexico, 2013, pp. 2414–2421.

[23] R. C. David, R. E. Precup, E. M. Petriu, M. B. Rădac，and S. Preitl, "Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity," *Information Sciences*, vol. 247, no. 15, pp. 154–173, Oct. 2013.

[24] Y. Liu, and H. Shang, "Improved Gravitational Search Algorithm for Grey Nonlinear Constrained Programming Problem," *Mathematics in Practice & Theory*, vol. 43, no. 7, pp. 99–103, Jan. 2013.

[25] B. Shawd, V. Mukherjee，and S. P. Ghoshal, "Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 55, no. 2, pp. 29–40, Feb. 2014.

[26] Y. Liu, and L. Ma, "Gravitational Search Algorithm for Minimum Ratio Traveling Salesman Problem," *Journal of Chinese Computer Systems*, vol. 34, no. 4, pp. 847–849, Apr. 2013.

[27] C. Li, H. Li, and P. Kou, "Piecewise function based gravitational search algorithm and its application on parameter identification of AVR system," *Neurocomputing*, vol. 124, no. 2, pp. 139–148, Jan. 2014.

[28] 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, May. 2013.

[29] S. Mirjalili, S. Z. M. Hashim, "New Hybrid PSOGSA Algorithm for Function Optimization," *in Proc. 2010 International Conference on Computer and Information Application (ICCIA)* , Tianjin, China, 2010, pp. 374–377.

Jie-sheng Wang received his B. Sc. And M. Sc. degrees in control science from University of Science and Technology Liaoning, China in 1999 and 2002, respectively, and his Ph. D. degree in control science from Dalian University of Technology, China in 2006. He is currently a professor and Master's Supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interest is modeling of complex industry process, intelligent control and Computer integrated manufacturing.

Jiang-Di Song is received her B. Sc. degree from University of Science and Technology Liaoning in 2012. She is currently a master student in School of Electronic and Information Engineering, University of Science and Technology Liaoning, China. Her main research interest is modeling methods of complex process and intelligent optimization algorithms.