Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **1**, January **2018**

Gravitational Search Algorithm with Dynamic Learning Strategy

Yan-Jiao Wang, Huan-huan Tao and Xiao-Jie Li

College of Information Engineering
Northeast Electric Power University
No.169, Changchun Rd., Chuanying, 132012, Jilin, China
lxj-635967767@163.com, taociguan@126.com, 635967767@qq.com

Received July, 2016; revised October, 2017

Abstract. *The gravitational search algorithm has been extensively applied in various fields, such as grid optimization, power system economic dispatching, pipeline scheduling, data mining and others; however, it complex engineering optimization problems prone to premature convergence. This paper proposes a gravitational search algorithm with the dynamic learning strategy (DGSA). An adaptive function is used as an alternative to the fixed value method to speed up convergence, and gravitation is calculated through a linear formula whereby individuals with high quality are selected iteratively to prevent falling into local optima and enhance diversity. Simulation results on six benchmark functions showed that the DGSA performs very well at avoiding premature convergence. Compared to GSA and other improved GSAs, the new algorithm likewise performs well in terms of not only convergence rate but also convergence precision.*
Keywords: Gravitational search algorithm, Linear formula, Function optimization

1. **Introduction.** The gravitational search algorithm (GSA) is an innovative heuristic algorithm based on the laws of gravity and motion, which were introduced recently by Rashedi and Nezamabadi-pour. The general objective of the algorithm is using Newton's law of gravitation and movement for the exchange of information [1]. Several studies have shown that GSA function optimization performance has the best possible effect compared to similar algorithms, e.g., the excellent particle swarm optimization algorithm, differential evolution algorithm, and others; as such, it has garnered significant research attention and research on optimization problems becomes one of the hottest topics of intelligent computation[2, 3]. Researchers developed a hybrid gravitational search algorithm in an effort to solve the economic dispatch problem power systems, for example, by combining OBL technology and GSA [4]. In another study, scholars used GSA to optimize the pipeline scheduling problem [5]. A novel gravitational search based kernel clustering technique was developed and applied to vibration fault diagnosis of hydro-turbine generating units [6]. The algorithm was proven well capable of clustering around the faulty samples of the units effectively and diagnosing different types of faults accurately. The GSA algorithm allows these applications to obtain better results, but the algorithm itself makes it easy to fall into local optima slows convergence speed, and introduces other important shortcomings. To this effect, many recent scholars have focused on improving the GSA algorithm to make it more directly applicable.

The gravitational particle swarm algorithm (PSOGSA), a combination of the characteristics of PSO and GSA, converges more quickly than the traditional GSA [7]. Recent researchers inspired by astronomical phenomena, proposed an improved GSA based on the black hole mechanism and verified its effectiveness by solving unimodal problems [8]. In another study, an opposition-based gravitational search algorithm (OGSA) was developed to avoid premature convergence [9]. These methods do, to a certain extent, improve the overall performance of the algorithm, but without fundamentally eliminating its inherent drawback. In this study, we utilized a dynamic adaptive function α to replace the static set value α to improve the GSA convergence rate. In order to balance global exploration and local search capacity while escaping local optimal, we also included a linear variation to establish the gravitational weight formula. We evaluated the proposed gravitational search algorithm with the dynamic learning strategy (DGSA) on six benchmark functions and compared it against the standard GSA, OGSA and PSOGSA. The results showed that DGSA performed better than the other algorithms in most of the test functions, effectively improving the performance of the GSA algorithm.

2. **Gravitational Search Algorithm.** In the GSA, the population of particles by the action of gravity of the mutual movement, due to the action of gravity causes the particles to move toward the most massive particle, and the quality's largest particle in the best position, the position information corresponding to the optimal solution of the optimization problem. GSA by gravitational interactions between the particles to optimize information sharing, guiding groups to expand the search area the optimal solution, the final movement of the particles of the optimal regional solutions.

To describe GSA in more detail, consider a D-dimensional space with N searcher agents in which the position of the ith agent is defined as follows:

$$x_i = \left(x_i^1, \cdots, x_i^d, \cdots x_i^D\right), i = 1, 2, \cdots N$$

The position of the ith particle is denoted by x_i^d. The quality of particles is determined by evaluating the fitness function values of the particles as follows:

$$\begin{cases} q_i\left(t\right) = \dfrac{fit_i\left(t\right) - worst\left(t\right)}{best\left(t\right) - worst\left(t\right)} \\ M_i\left(t\right) = q_i\left(t\right) \Bigg/ \displaystyle\sum_{j=1}^{N} q_j\left(t\right) \end{cases} \tag{1}$$

Where $M_i\left(t\right)$ represent the mass. $Fit_i\left(t\right)$ represents the fitness value of the ith particle in iteration of the t. $Best\left(t\right)$ represents the best fitness value in iteration t and $worst\left(t\right)$ represents the worst fitness value in iteration t.

$$best\left(t\right) = \min_{i \in \{1,2,\cdots,N\}} fit_i(t) \tag{2}$$

$$worst\left(t\right) = \max_{i \in \{1,2,\cdots,N\}} fit_i\left(t\right) \tag{3}$$

Total forces applied on an agent from a set of heavier masses should be considered based on the law of gravity as stated in (4).

$$F_{ij}^d\left(t\right) = G\left(t\right) \frac{M_{pi}\left(t\right) M_{aj}\left(t\right)}{R_{ij}\left(t\right) + \varepsilon} \left(x_j^d\left(t\right) - x_i^d\left(t\right)\right) \tag{4}$$

Where $M_{aj}\left(t\right)$ is the active gravitational mass related to agent j, $M_{pi}\left(t\right)$ is the passive gravitational mass related to agent i, $G\left(t\right)$ is gravitational constant at time t, ε is a very small value used in order to escape from division by zero error whenever the Euclidean distance between two agents i and j is equal to zero, and $R_{ij}(t)$ is the Euclidian distance between two agents i and j in (5)

$$R_{ij}(t) = \|(X_i(t)), (X_j(t))\|_2 \tag{5}$$

In GSA, the gravitational constant G will take an initial value G_0, and it will be reduced with time as given in (6)

$$G(t) = G_0 \times e^{-\alpha t/T} \tag{6}$$

Where G_0 is a gravitational constant, usually 100. α is 10, t and T are the current and the total number of iterations (the total age of the system), respectively.

In a problem space with dimension equal to d, the total force that acts on agent i is calculated by the following equation:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \tag{7}$$

Where $rand_j$ is a random number in the interval [0,1]. The random component has been included in this formula to have a random movement step along the gravitational force of each agent and the final resultant force. This helps to have more diverse behaviors in moving the search agents.

Newton's law of motion has also been utilized in this algorithm, which states that the acceleration of a mass is proportional to the applied force and inverse to its mass, so the accelerations of all agents are calculated as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{8}$$

Where t is a specific time and M_i is the mass of object i.

The velocity and position of agents are calculated as follows:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{9}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{10}$$

Where $rand_j$ is a random number in the interval [0,1].

The flowchart of gravitational search algorithm is shown in Figure. 1.

3. **Improved Gravitational Search Algorithm.**

At present, many intelligent optimization algorithms including GSA suffer slow convergence and readily fall into local optima [10]. We developed the DGSA technique in an effort to eliminate these problems.

3.1. **Hyperbolic Tangent function dynamic adaptive α.**

Of course, the parameter settings of the algorithm are crucial in terms of optimizing performance. In the GSA, there are two fixed parameters: G_0 and α. Under Rashedi settings, parameter G_0 value is 100 and α value is 20. The parameter α determines the global exploration and local search development [9]. During the algorithm, α values should be small initially in order to ensure the growth of the individual steps and to prevent falling into local optima. Later in the algorithm, the values should become larger in order to accelerate the convergence speed, with a relatively small step toward the optimal solution to ensure high accuracy. In short, the α value should be increased algebraically at the appropriate iteration.

We identified a hyperbolic tangent function in which output increases nonlinearly as the number of iterations increased. It is feasible to use a hyperbolic tangent function as α and to control G.

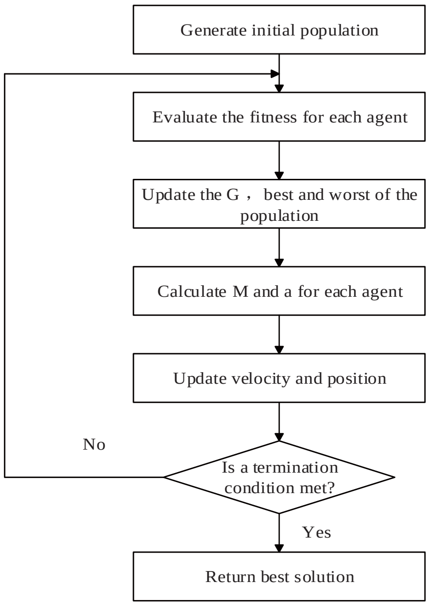

FIGURE 1. The flowchart of GSA algorithm

The attenuation factor function is defined by a hyperbolic function as follows:

$$\alpha(t) = \rho * \tanh(\lambda * (\frac{t}{T} - 0.5)) + \kappa \tag{11}$$

Where t is the current iteration and T is the maximum number of iterations; ρ, λ, and κ are the parameters of the α function. We selected suitable parameters to control the variation range of α and set ρ, λ, and κ to 0.2, 10, and 20, respectively. The dynamic α produced a smaller G in early iterations, helping GSA perform exploration, and a larger G in later iterations, helping the GSA perform exploitation.

3.2. **Gravity Improvement.** The global exploration and local search have an important influence on the performance of the optimization algorithm. Equations (1)-(8) can be substituted into Eq. (9) to obtain the following formula:

$$x_i^{t+1} = x_i^t + rand * v_i^t + \sum_{j=1}^{N}\left(G_0 \times e^{-\alpha t/T} * \frac{M_j^t}{R_{ij}}\left(rand * \left(x_j^t - x_i^t\right)\right)\right) \tag{12}$$

As evidenced by Eq. (11), the GSA algorithm is similar to the mutation operator in the differential evolution algorithm [11]. The latter half of the formula is the product of the difference vector, mass, and distance between the individual and other individuals. The prediction formula of unrated knowledge is as follows after the time function is introduced:

$$L = \sum_{j=1}^{N}\left(G_0 \times e^{-\alpha t/T} * \frac{M_j^t}{R_{ij}}\left(rand * \left(x_j^t - x_i^t\right)\right)\right) \tag{13}$$

Equation (13) is the result of learning from the individual to the others. The quality and the front part of the multiplication result can be utilized to determine the proportion of the overall learning; this can be regarded as the overall difference in front of the weight, which has a certain algebraic relation with the fitness value. To efficiently algebraically calculate the function, we developed the following linear formula.

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) * (1 - t/T) \tag{14}$$

Where t is the current iteration and T is the maximum number of iterations. $\omega_{\max}$ and $\omega_{\min}$ were set to 0.9 and 0.7.

Equation (13) can be rewritten as follows:

$$F_i^d(t) = G(t) * \sum_{j=1}^{N} \left(\omega * \frac{M_j^t}{R_{ij}} \left(rand * \left(x_j^t - x_i^t \right) \right) \right) \tag{15}$$

Via the linear weighting formula, individuals explore larger regions in the initial iteration, and then quickly locate the approximate location of the optimal solution; ω decreases as the number of iterations increases, then the individuals slow down and begin finer, local search. The most prominent feature of GSA is the entire population, which must rely on gravity to optimize between individuals. Gravity is equivalent to an information transfer tools optimizations according to the characteristics of the algorithm, the finer the individual quality, the finer the gravitational force. Under the action of gravity, the whole population moves toward the quality of the largest direction to search the optimal solution for the problem. The gravitational force is calculated as the first individual is added to the last one. In the early iterations, due to the diversity of individual populations, the population can quickly find optimal solutions. Population diversity is decreased in later iterations, which can easily reduce accuracy and fall into the local optimal solution. The quality of the individual is divided by quality into three categories: excellent, mid-level, and poor in order jump local optima. At the beginning of the algorithm, the population is given high probability and the individuals involved in the operation are given the highest quality possible to strengthen the exchange of information between them to enhance the population's mining capacity, and to promote population convergence. In subsequent iterations, it is necessary to enhance the local search ability to the individual, ensure mid-level quality, and assign moderate probability to the population; this creates poor individuals with relatively good individual movement able to conduct a fine-tuned search. A smaller portion of a given population with high quality should be selected to altogether avoid local optima; this balances the overall exploration and local development capabilities of the algorithm. The pseudo code of gravity operator can be seen in following:

3.3. **Step-wise DGSA Algorithm.** The reactive power optimization based on DGSA takes place in the following steps.

Step1: Population initialization

Step2: Fitness evaluations of the agents

Step3: Update $M_i(t)$ based on Eq. (1)

Step4: Calculating $\alpha(t)$ based on Eq. (11) and updating $G(t)$based on Eq. (6)

Step5: Calculating the total forces in different directions using Eq. (15)

Step7: Calculating the acceleration $a_i^d(t)$ by Eq. (8) and updating velocity $v_i^d(t)$ and position $x_i^d(t)$ by Eq. (9) and (10), where $i = 1, \cdots N$

Step8: Repeat Steps 2-7 until stopping termination condition is met.

4. **Experiments and Results.** We ran simulation experiments to evaluate the performance of the algorithm and the effectiveness of the proposed technique. We applied the proposed algorithm and a few others to six standard benchmark functions for the sake of comparison [12, 13, 14, 15]. Table 1 describes the benchmark functions, the ranges of their search space, and their optimal values. $F_1 \sim F_4$ are unimodal. The convergence rates of the functions increased to a greater extent than the final optimization results. $F_5 \sim F_6$ are multimodal. Having many local minima, the algorithm must be capable of finding the optimum solution without falling into local optima.

if rand $< P$ && $t < T$ **then**
  for $i = 1 : N$ **do**
    for $j = 1$: *kbest* **do**
      calculating the total forces in different directions using Eq. (15)
    end for
  end for
else
  if rand$< P'$ **then**
    for $i = 1 : N$ **do**
      for $j = 1$: *kbest1* **do**
        calculating the total forces in different directions using Eq. (15)
      end for
    end for
  else
    for $i = 1 : N$ **do**
      for $j = 1$: *kbest2* **do**
        calculating the total forces in different directions using Eq. (15)
      end for
    end for
  end if
end if
Kbest, *kbest1*, and *kbest2* stand for excellent, mid-level, and poor particle mass.

Table 1. Test Function

Function	Range	Opt
$F_1(X) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0
$F_2(X) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$	0
$F_3(X) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^D$	0
$F_4(X) = \sum_{i=1}^{D} ([x_i + 0.5])^2$	$[-100, 100]^D$	0
$F_5(X) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500, 500]^D$	0
$F_6(X) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	0

4.1. **Set Experiment Parameters.** We set the same initial parameters for the tests we ran on GSA, OGSA, PSOGSA, and DGSA: population size=50; dimension = 30, and maximum iterations =1000.

4.2. **Simulation Results and Analysis.** The performance of the DGSA algorithm was compared against the standard GSA, OGSA, PSOGSA, and improved GSA in terms of average, minimum, maximum, and standard deviation of best-so-far solutions. The results are listed in Table 2.

The algorithm with smaller average is stronger for searching the optimal solution. And the algorithm with lower variance has higher stability. The results are shown in Table

TABLE 2. The Result Of Test Function

Function	Algorithm	Minimum	Average	Maximum	Std. dev
F_1	GSA	1.21e-017	2.18e-017	3.90e-017	5.82e-018
	OGSA	6.96e-017	1.06e-017	1.53e-16	2.37e-017
	PSOGSA	1.30e-019	2.51e-019	3.43e-019	5.78e-020
	DGSA	4.88e-031	9.41e-024	1.86e-022	4.15e-023
F_2	GSA	1.70e-008	2.38e-008	3.12e-008	4.43e-009
	OGSA	3.88e-008	4.56e-008	5.53e-008	4.17e-009
	PSOGSA	1.78e-009	7.8479	118.6478	26.8984
	DGSA	1.06e-015	2.85e-015	6.10e-015	1.38e-015
F_3	GSA	1.05e-095	6.83e-087	8.83e-088	2.14e-088
	OGSA	1.14e-105	1.90e-101	3.78e-100	8.49e-101
	PSOGSA	23.687	6.63e+004	1.51e+004	4.91e+003
	DGSA	1.21e-133	4.70e-121	9.40e-120	2.10e-120
F_4	GSA	1.22e-017	2.25e-017	3.7e-017	5.83e-018
	OGSA	7.69e-017	1.10e-016	1.43e-016	1.69e-017
	PSOGSA	1.61e-019	505.01	1.01+004	2.26e+003
	DGSA	4.07e-031	1.58e-024	3.15e-023	7.03e-024
F_5	GSA	-3.59e+003	-2.86e+003	-2.16e+003	430.7376
	OGSA	-4.12e+003	-2.91e+003	-2.08e+003	581.3082
	PSOGSA	-3.30+003	-2.43e+003	-1.77e+003	419.34
	DGSA	-9.25e+003	-7.74e+003	-5.92e+003	853.2039
F_6	GSA	21.0592	39.2327	73.8426	12.5663
	OGSA	20.82	41.07	62.70	11.36
	PSOGSA	3.33e-016	9.090	90.7526	27.9244
	DGSA	0	4.93e-004	0.0049	0.0015

2. In almost all test functions, both the accuracy stability and convergence of the DGSA algorithm were optimal in terms of solving unimodal and multimodal functions.

For a more direct comparison of various algorithm's convergence rates, we drew curves of the convergence process as shown in Figure 2 and Figure 7, where the horizontal axis represents the number of iterations of the population and the ordinate axis represents the number of iterations corresponding to the optimal solution of the population.

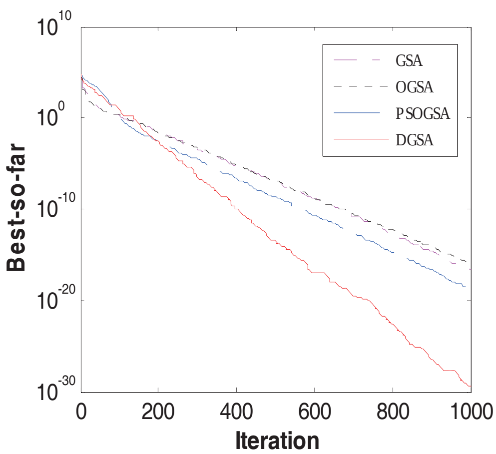

FIGURE 2. F_1

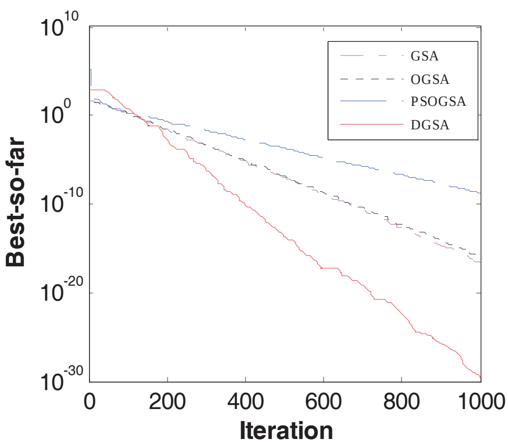

FIGURE 3. F_2

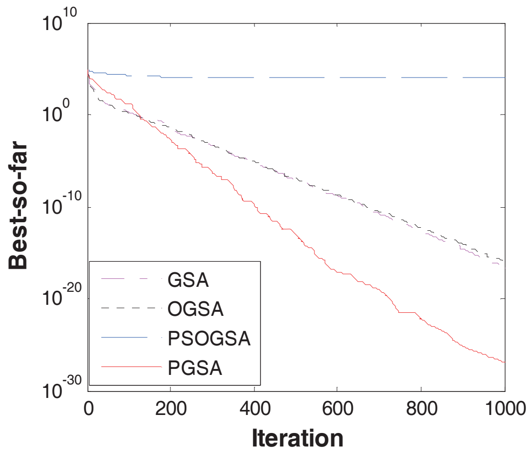

FIGURE 4. F_3

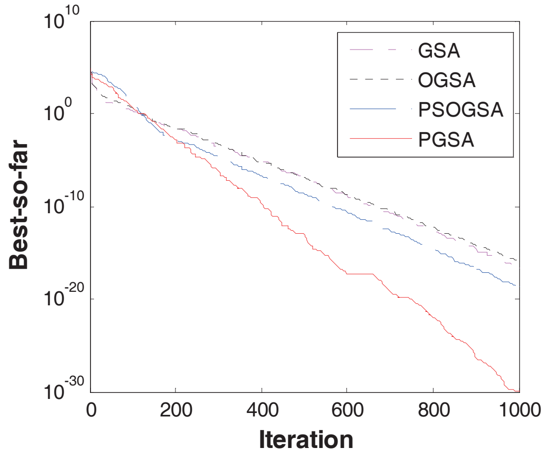

FIGURE 5. F_4

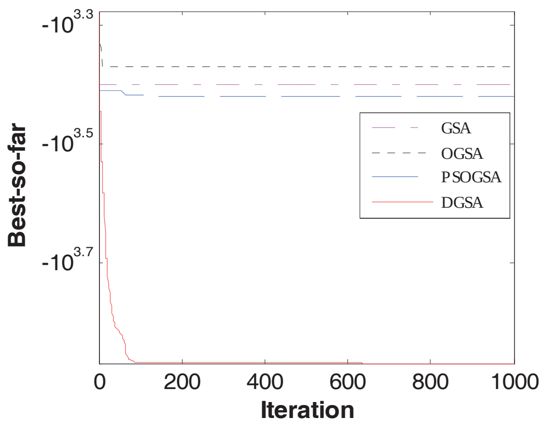

FIGURE 6. F_5

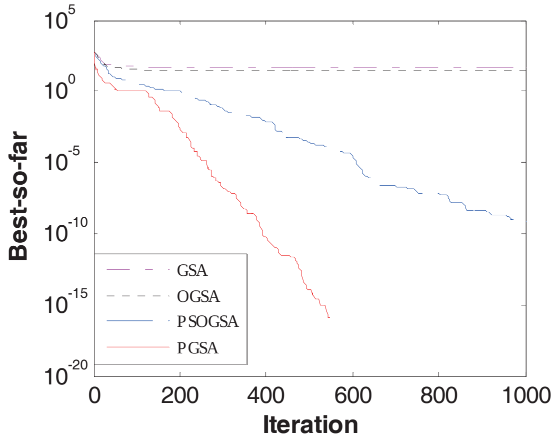

FIGURE 7. F_6

The simulation results showed that the convergence speed and accuracy of the DGSA algorithm were significantly better than the standard GSA, OGSA, and PSOGSA algorithms. In effect, the improvement measures we designed were effective; the dynamic learning strategies effectively balance the global exploring ability and local search ability while avoiding falling into local optima.

5. **Conclusions.** This paper presents a gravitational novel search algorithm with a dynamic learning strategy. The algorithm has two particularly notable characteristics: first, the static dynamic adaptive function $\alpha(t)$ is replaced to set the value of α to accelerate the convergence rate; second, gravity is calculated in different iterations according to the size of the population and individual quality. In this way, population diversity is effectively maintained during optimization. We tested ours and other similar algorithms on six standard benchmark functions for the purposes of comparison and found that the proposed algorithm yields better solutions with faster convergence.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant NO.61501107 and the Education Department of Jilin province science and technology research project of "13th Five-Year".

REFERENCES

[1] R. E, N H, and Saryazdi S. A gravitational search algorithm, *Information sciences*, vol.179, no.13, pp. 2232-2248, June 2009.

[2] LJ. Sun, PL. Du, F. Xiao. Coverage-Enhancing Algorithm for Directional Sensor Networks Based on Particle Swarm Optimization, *Journal of Computer Research and Development*, vol.47, pp.22-25, Sep 2010.

[3] D Cai. A New Evolutionary Algorithm Based on Uniform and Contraction for Many-objective Optimization, Journal of Network Intelligence, vol. 2, no. 1, pp. 171-185, Feb 2017.

[4] B.S, V. M, and SP. Ghosal. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. *Electrical Power and Energy Systems*, vol.35, no.1, 2012.

[5] D, M, and Guvenc U. Determination of the PID controller parameters for speed and position control of DC motor using gravitational search algorithm, *The 7th International Conference on Electrical and Electronics Engineering*, Bursa, IEEE Press, pp.225-229, 2011.

[6] Li, Zhou, and Xiao jian. Vibration Fault Diagnosis of Hydroelectric Generating Unit Using Gravitational Search Based Kernel Clustering Method, *Proceedings of the CSEE*, vol.2, no.33, pp. 98-104, 2013.

[7] Tsai, T, Wu Y W, et al. Gravitational particle swarm, *Applied Mathematics and Computation*, vol. 219, no. 17, pp. 9106-9117, 2013.

[8] D, Nezamabadi-pour H. A hybrid algorithm based on gravitational search algorithm for unimodal optimization ,*2012 2nd international conference on computer and knowledge engineering* (ICCKE), 2012, pp. 129-132

[9] B, V. M, S.P. and Ghoshal C. Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm, *Electrical Power and Energy Systems*, pp. 29-40, 2014

[10] K.M, Taha M R, El-Shafie A, et al. A modified gravitational search algorithm for slope stability analysis, *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp.1589-1597, 2012.

[11] Jiang, Tan, Dong Li-ming, et al. Application of modified gravitational search algorithm in slope stability analysis, *Chinese Journal of Geotechnical Engineering*, vol. 38, no. 3, pp. 419-425, 2016.

[12] F. Weifeng. *Analysis and improvement of the gravitational search algorithm*, Master Dissertation Guangzhou: Guangdong University of Technology, 2014.

[13] R. S, Kenneth Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[14] C, Gong W Y, Ling Charles-X. Research on a novel biogeography-based optimization algorithm based on evolutionary programming, *Systems Engineering Theory & Practice*, vol. 30, no. 6, pp. 1106-1112, 2010.

[15] Saucer T W, S V. Optimizing nanophotonic cavity designs with the gravitational search algorithm, *Optics Express*, vol.21, no.18, pp.20831-20836, 2013.