

Study on Model and Optimization Algorithm of Multi-Vehicle and Multi-Depot Emergency Vehicle Dispatch Problem

Yang Wang^{1,a}, Fan Lin^{2,b,*}, Tiejun Wang^{3,c} and Kaijun Wu^{1,d}

¹*School of Electronic and Information Engineering, Lanzhou JiaoTong University, Lanzhou 730070, China*

²*Software School, Xiamen University, 361000, Xiamen, China*

³*Department of mathematics and computer science, Northwest University for Nationalities, Lanzhou 730030, China*

^a597552325@qq.com, ^{b*}iamafan@xmu.edu.cn, ^cwtj@mail.lzjtu.cn,

^dwkj@mail.lzjtu.cn

Abstract

The multi-type vehicle and multi-depot emergency vehicle dispatch is a typical problem in emergency scheduling, and it is a NP puzzle. Under the constraints of transportation cost and carrying load, etc., a mathematical model concerning this problem was constructed in present study; and then it was solved by an improved gravitational search algorithm (GSA). Because the standard GSA is easily trapped in local optimum and appears “precocious phenomenon”, an improved GSA was proposed to modify gravity coefficient and velocity in selection formula, and to choose the updated particle’s positions according to the principle of survival of the fittest. The simulation results show that this proposed algorithm is feasible and effective, and is superior to differential evolution algorithm and the standard GSA in solving the problem of multi-type vehicle and multi-depot emergency vehicle dispatch.

Keywords: *Multi-vehicle and Multi-depot; emergency vehicle dispatch problem; gravity coefficient; velocity selection formula; survival of the fittest rule*

1. Introduction

China has a broad land, so the occurrence probability of all kinds of natural disasters is increased. In recent years, the frequent occurrences of natural disasters such as tsunami, earthquake and snow storm have caused great damages to people's life and production. Therefore, how to schedule relief vehicles to multiple disaster sites quickly and efficiently becomes a very important problem. Multi-vehicle and Multi-depot emergency vehicle dispatch problem means: under the conditions of many relief points, a variety of vehicles and a plurality of disaster points taking part in the distribution of sources, how select vehicles, how organize transport, to make the least mileage and the lowest cost of transport. In recent years, the algorithms such as divide-and-conquer algorithm and genetic algorithm have provided preliminary methods to solve this type of problem. For instance, Wang Lingling used heuristic algorithm to program scheduling task of vehicles, and then designed genetic algorithm on remaining disaster points to adjust distribution tasks of each logistics distribution center and delivery order of each car, finally getting vehicle’s driving route[1]; GanYong introduced a multi-depot and multi-resource scheduling model, which is based on the urgency degree of resource requirements, and he used divide-and-conquer algorithm to get a scheduling plan which has the highest composite benefits[2]; In the study of urban public emergency resource allocation, Fan Wenjing designed a two-stage algorithm(improved genetic algorithm and linear

* Corresponding Author

programming)[3]. But because these algorithms have some shortcomings, *e.g.* long searching time, immature convergence, therefore the convergence speed and accuracy need further improvement.

Gravitational search algorithm (GSA) [4] is a new optimized search algorithm inspired by the mind of universal gravitation, and it was proposed by professor Esmat Rashedi from the University of Kerman in 2009. It's similar to particle swarm optimization (PSO), and is a heuristic algorithm. It tries to find optimal regions of complex searching space through the gravitational interaction between particles in population. Existing studies demonstrate that the convergence properties of GSA is obviously better than PSO, genetic algorithm (GA) and other bionic intelligent algorithms [4]. Some experts and scholars put forward improved algorithms, such as Li Pengfei improved overall system performance of GSA by combining two optimization operators in flatness pattern recognition [5]. But the improved algorithms also have some disadvantages, *e.g.* premature convergence, poor ability in local optimization and the lack of effective acceleration mechanism. For this reason, this paper proposes an improved gravitational search algorithm (IGSA). The experimental results show that the improved algorithm can solve the multi-vehicle and multi-depot emergency vehicle scheduling problem effectively, and it is superior to the basic GSA. Therefore, this paper provides a more reasonable and efficient method to solve this multi-vehicle and multi-depot emergency vehicle scheduling problem.

2. The Mathematic Model of Multi-Depot Emergency Vehicle Scheduling Problem

Multi-vehicle and Multi-depot emergency vehicle scheduling problem can be generally described as: M ($i = 1, 2, \dots, M$) relief points need to transport relief goods to N ($j = 1, 2, \dots, N$) disaster sites, and each relief point has K_h cars whose load is Q_h ($h = 1, 2, \dots, H$) tons. The unit distribution cost of Q_h tons car is C_h , the demand g_j of each disaster site is less than Q_h , and the transportation distance from relief point i to affected site j is d_{ij} . Assume any relief point can select any number and any type of vehicles from any relief point, and the vehicles must return to original depot after completing disaster relief task. Designing a set of reasonable scheduling scheme, while satisfies all demand of disaster sites, could minimize the total cost of distribution.

Assume the number of relief points are $1, \dots, M$, the number of affected sites are $1, \dots, N$, the definition of variables is shown as formula (1):

$$x_{ij}^{mnhk} = \begin{cases} 1 & \text{k h-model vehicles of relief point m drive from barn i to disaster site j} \\ 0 & \text{else} \end{cases} \quad (1)$$

Then we can get mathematical model of multi-vehicle and multi-depot emergency vehicle scheduling problem, as shown in formula(2):

$$f = \min \sum_{n=1}^N \sum_{h=1}^H \sum_{m=1}^M \sum_{k=1}^{K_h} \sum_{j=1}^N d_{ij} x_{ij}^{mnhk} C_h \quad (2)$$

The constraints set are shown as formula(3) to (6):

$$\sum_{i=1}^M \sum_{k=1}^{K_m} x_{ij}^{nmk} \leq K_h \quad (3)$$

$$\sum_{i=1}^M x_{ij}^{mnhk} = \sum_{j=1}^N x_{ij}^{mnhk} \leq 1, \quad m \in \{1, \dots, M\}, \quad h \in \{1, \dots, H\}, \quad k \in \{1, \dots, K_h\} \quad (4)$$

$$\sum_{i=1}^M g_i \sum_{j=1}^N x_{ij}^{mnhk} \leq Q_h, \quad m \in \{1, \dots, M\}, \quad h \in \{1, \dots, H\}, \quad k \in \{1, \dots, K_h\} \quad (5)$$

$$\sum_{j=1}^{N+M} \sum_{m=1}^M \sum_{k=1}^{K_h} \sum_{h=1}^H x_{ij}^{m hk} = 1, \quad i \in \{1, \dots, N\} \quad (6)$$

$$\sum_{i=1}^{N+M} \sum_{m=1}^M \sum_{k=1}^{K_h} \sum_{h=1}^H x_{ij}^{m hk} = 1, \quad j \in \{1, \dots, N\} \quad (7)$$

$$\sum_{i=1}^M x_{ij}^{m hk} = \sum_{j=1}^N x_{ij}^{m hk} = 0, \quad m \in \{1, \dots, M\}, \quad h \in \{1, \dots, H\}, \quad k \in \{1, \dots, K_h\} \quad (8)$$

In the model, formula(2) is objective function, minimizing the total cost of vehicle distribution; formula(3) represents the number of vehicles involved in disaster-relief task don't exceed the total number of each relief point; formula(4) ensures that every emergency vehicle will be able to return to original relief point; formula(5) represents the load of each car should not exceed its load limit; formula(6) and (7) ensure that each customer can be served only once by a specific car; formula(8) ensures the vehicles involved in disaster-relief task can't be driven from one relief point to another one.

3. The Basic Gravitational Search Algorithm (GSA)

The Gravitational search algorithm was proposed mainly based on the physics law of universal gravitation. GSA regards all particles as objects with mass, and they could make resistanceless motion. In solution space, each particle will be effected by the gravity of other particles, and produces acceleration pointing at particle of larger mass. Due to the mass of particles are related to it's fitness value, the larger the fitness value of the particle, the larger the mass of the particle. In the process that small mass particle moving to large mass particle, the algorithm can gradually approach the optimal solution. The difference between GSA and clustering algorithms such as ant colony algorithm is that particles don't have to perceive environment through environmental factors, but realize the optimization of information sharing by gravitational interaction between individuals. Therefore, the particle can perceive global situation, and launch a search for environment without the influence of environmental factors.

In GSA, each particle has four features: location, inertial mass, active gravitational mass and passive gravitational mass. The interaction force between particles lead the particle into moving towards massive particles, and particle's position is corresponding to it's fitness [4].

Assume that it is in a *n-dimensional* search space, and *N* particles form a initial group. Define the *ith* particle's position is:

$$X_i = (x_i^1, x_i^2, \dots, x_i^d, \dots, x_i^n), i = 1, 2, \dots, d, \dots, N. \quad (9)$$

Assume that at time *t*, the inertial mass $M_i(t)$ of particle *i* is:

$$m_i(t) = \frac{fitness_i(t) - worst(t)}{best(t) - worst(t)} \quad (10)$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (11)$$

Among them, $i = 1, 2, \dots, N$, $m_i(t)$ is a intermediate variable, $fitness_i(t)$ is fitness value of particle *i* at time *t*, $best(t)$ is the best fitness value of whole particle swarm at time *t*, and $worst(t)$ is the worst. Aiming at the problem of minimum optimization, the best and the worst fitness values are respectively represented as:

$$best(t) = \min_{j \in \{1, 2, \dots, N\}} fitness_j(t) \quad (12)$$

$$worst(t) = \max_{j \in \{1,2,\dots,N\}} fitness_j(t) \quad (13)$$

According to Newton's law of gravitation, at time t , in d -dimensional space, the force of the j th particle to the i th particle is:

$$F_{ij}^d(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (14)$$

In formula, $x_i^d(t)$ means the i th particle's position in d -dimensional space at time t ; $x_j^d(t)$ means the j th particle's position in d -dimensional space at time t ; ε is a very small positive number, $R_{ij}(t)$ is Euclidean distance between particle i and particle j , $G(t)$ is gravitation coefficient at time t . The formula is shown as the following:

$$R_{ij}(t) = \|x_i(t), x_j(t)\|_2 \quad (15)$$

$$G(t) = G_0 e^{-\beta \frac{t}{T}} \quad (16)$$

GSA belongs to natural intelligence algorithm. Among them, G_0 and β are constants, T is the maximum number of interaction.

In GSA, it is assumed that the resultant force $F_i^d(t)$ acting on i th particle equals to occlusal force of all other particles in d -dimension, at time t :

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (17)$$

Among them, $rand_j$ is a random number between $[0, 1]$.

According to Newton's second law, in d -dimensional space, at time t , the acceleration of particle i is defined as the following:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (18)$$

Finally, the particle i updates velocity and position according to formula(19) and (20):

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (19)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (20)$$

Among them, $v_i^d(t)$ means the velocity of particle i in d -dimensional space, at time t .

4. The Improved Gravity Search Algorithm (IGSA)

4.1. The Improvement of Gravity Coefficient

The gravity coefficient $G(t)$ is an important parameter for GSA, at the beginning of optimization, GSA needs larger $G(t)$ leading population to explore optimal solution areas quickly; but in the late stage of optimization, it requires smaller $G(t)$ to perform local search in explored solution area. However, in standard GSA, $G(t)$ is exponent function, which falls fast, and leads the global search ability of GSA into ebbing fast, thereby reducing global convergence speed.

In order to make GSA explore the optimal solution area faster and more accurately, linear function is used to improve the gravity coefficient $G(t)$. The formula is shown as the following: $G(t) = G_0(1 - t/T)$ (21)

Among them, t is current number of iterations. T is the largest number of iterations. For it's difficult to find a suitable constant G_0 , dynamic adjustment is used into G_0 . The formula is shown as the following:

$$G_0 = \gamma \max_{d \in \{1, \dots, n\}} (|x_{\max}^d - x_{\min}^d|) \quad (22)$$

Among them, γ is a constant. After several tests, we set it to 2.5 in the paper. x_{\min}^d means the minimum location value of individual in current population, d -dimensional space, and x_{\max}^d means the maximum.

4.2. The Improvement of Velocity Equation

GSA and PSO are two kinds of optimization algorithms, and their solutions are all obtained by moving individuals in search space. In the search process of GSA, the moving direction of individuals are obtained by calculating resultant force of all other particles. In the link of updating location, only the current position of individual works, without group memory function. The basic PSO algorithm [6] not only considers particle's own experience, but also includes group information sharing between particles. The improved equation of particle swarm algorithm is:

$$v_i^d(t+1) = wv_i^d(t) + c_1 \text{rand}_j (p_{\text{best}}^d - x_i^d(t)) + c_2 (g_{\text{best}}^d - x_i^d(t)) \quad (23)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (24)$$

In formula, w is inertia weight; rand_j , are random variables between $[0, 1]$; c_1, c_2 are constants between $[0, 1]$; $p_{\text{best}}^d(t)$ is the best position that particle i has experienced; $g_{\text{best}}^d(t)$ is the best position that all particles have experienced.

The memory and group communication functions based on PSO is introduced to improve GSA. According to the new strategy, the improved space search method abides by the laws of motion, and adds memory and group communication functions. The new equation is defined as the following:

$$v_i^d(t+1) = \text{rand}_i v_i^d(t) + a_i^d(t) + c_1 \text{rand}_j (p_{\text{best}}^d - x_i^d(t)) + c_2 \text{rand}_k (g_{\text{best}}^d - x_i^d(t)) \quad (25)$$

In formula, $\text{rand}_j, \text{rand}_k, \text{rand}_i$ are random variables between $[0, 1]$; c_1, c_2 are constants between $[0, 1]$; $p_{\text{best}}^d(t)$ represents the best position that particle i has experienced; $g_{\text{best}}^d(t)$ represents the best position that all particles have experienced. By adjusting the value of c_1, c_2 , we can balance the influence of gravity, memory and social information on search process.

4.3. The Improvement of Position Selection Formula

A "greedy" selection mode [7], which is similar to select operation of differential evolution, is used in the paper. Namely, the updated new individual is accepted by population if and only if the fitness value of new vector individual is better than the fitness value of target vector individual. Otherwise, the individual of last generation will be retained in next generation. Its selection operation is shown as the following:

$$x_i^d(t+1) = \begin{cases} \text{new}_i^d(t+1) & f(\text{new}_i^d(t+1)) < f(x_i^d(t)) \\ x_i^d(t) & \text{otherwise} \end{cases} \quad (26)$$

In each generation, the updated new individual can replace the location of current individual only when its fitness value is less than the previous one. By using this method, we can realize the survival of the fittest, and the population will evolve towards the optimal solution.

5. The Application of Improved Gravity Search Algorithm to Multi-Vehicle and Multi-Depot Emergency Vehicle Dispatch Problem

5.1. Coding Strategy of Particle

One of the key problems for realizing algorithm is how to find a suitable representation method to make the individual corresponding to the solution. This paper expresses vehicle routing optimization problem with 4 dimensional vector X . The first dimension X_r represents the information of the relief points participating in rescue; the second dimension X_s represents the serial number information of the cars taking part in rescue; the third dimension X_u represents the model information of the vehicles involved in rescue; the fourth dimension X_t represents the driving distance of each car.

5.2. Decoding Strategy of Particles

In order to change the travel path order of car i , we have to adjust the X_t . According to the order of elements in vector X_t , adjustment function Af can be determined. Firstly, we could find the disaster site i the car serving to; then sort them from the smallest to the largest according to the size of corresponding X_t ; lastly, determine the driving route order of vehicle i . For example, suppose there are 10 disaster points, 2 car depots, 2 kinds of car (A and B), and the number of each kind of car are 3. The position vector X of one particle is shown as the Table 1, and the position vector X which has been adjusted is shown as the Table 2.

Then the corresponding solution path is:

Relief point 1: Vehicle1, Modle A: 6→1

Vehicle2, Modle B: 8→4

Relief point 2: Vehicle3, Modle A: 3

Vehicle4, Modle A: 2→5→7

Vehicle5, Modle A: 9→10

The coding and decoding strategy of the particle can ensure the service to all disaster sites, and the service can only be done by a specific car, which makes the calculation more simple.

Table 1. Vector X before Adjusting

Customer	1	2	3	4	5	6	7	8	9	10
X_r	1	2	2	1	2	1	2	1	2	2
X_s	1	4	3	2	4	1	4	2	5	5
X_u	A	B	A	B	B	A	B	B	A	A
X_t	0.3	3.2	2.6	1.3	4.3	0.1	4.6	0.6	5.8	5.9

Table 2. Vector X after Adjusting

Customer	1	2	3	4	5	6	7	8	9	10
X_r	1	2	2	1	2	1	2	1	2	2
X_s	1	4	3	2	4	1	4	2	5	5
X_u	A	B	A	B	B	A	B	B	A	A
X_t	2	1	1	2	2	1	3	1	1	2

5.3. The Steps of IGSA

Step1. Set algorithm parameters: population size N , gravity coefficient G_0 , parameters ε , β , the maximum τ_{max} and the minimum τ_{min} of inertia weight, the maximum number of iterations, and make $t=0$;

Step2. Initialize the population;

Step3. Caculate the fitness value of each particle;

Step4. Caculate the inertial mass of each particle according to formula(10) and (11);

Step5. Update the gravity coefficient function according to formula(21) and (22);

Step6. Caculate the resultant force acting on each particle according to formula(17);

Step7. Caculate the acceleration of each particle according to formula(18);

Step8. Update the velocity of each particle according to formula(24);

Step9. Update the position of each particle according to formula(25) ;

Step10. Verify whether the algorithm satisfies the terminate condition, if meets, the algorithm ends, then output the optimal solution; otherwise, make $t=t+1$, and return to step3.

6. Example Analysis

There is a vehicle scheduling problem which has 3 relief points and 25 disaster sites. The location(unit: km) and the demand(unit: ton) of each disaster point are shown in Table 3. The location(unit: km) of each relief point, vehicle model, the number of each model, load capacity(unit: ton) and the unit distribution cost(RMB/km) are shown in Table 4.

Table 3. The Information of Disaster Point

disaster site	location	demand	disaster site	location	demand
1	(12, 14)	6	14	(66, 32)	3
2	(95, 86)	6	15	(64, 77)	3
3	(41, 97)	1	16	(83, 4)	7
4	(90, 13)	8	17	(19, 46)	4
5	(27, 99)	6	18	(78, 96)	9
6	(70, 52)	7	19	(42, 36)	4
7	(53, 40)	5	20	(7, 78)	5
8	(58, 3)	4	21	(68, 24)	6
9	(6, 56)	7	22	(11, 23)	4
10	(21, 28)	2	23	(82, 45)	2
11	(32, 8)	3	24	(32, 82)	4
12	(45, 74)	8	25	(35, 54)	3
13	(94, 42)	6			

Table 4. Information of Relief Point

relief point	location	model	load	number	unit distribution cost
26	(45, 16)	A	20	2	5
		B	15	2	4
27	(82, 75)	A	20	2	5
		B	15	3	4
28	(24, 65)	A	20	3	5
		B	15	3	4

This study uses basic GSA and IGSA to solve the problem. After calculation, the optimal scheduling scheme is 3 relief points and 7 relief vehicles participating in relief task. Among them, relief point, vehicle model, driving route, driving distance(unit: km) and the total cost of transportation are shown in Table 5. Figure 1 illustrates a curve that the suitable value(total cost of transportation) will vary based on the number of iteration.

Table 5. Distribution Data Information of Each Car

Relief point	model	Driving route	distance	cost
26	B	26→21→14→7→26	99.3	397.3
26	A	26→4→16→8→26	99.9	499.5
26	A	26→19→10→22→1→11→26	99.1	495.4
27	B	27→13→23→6→27	87.3	349.3
27	A	27→2→18→15→27	78.5	392.3
28	A	28→12→3→5→28	96.8	484.2
28	A	28→20→9→17→25→28	93.3	466.4
合计			654.2	3084.4

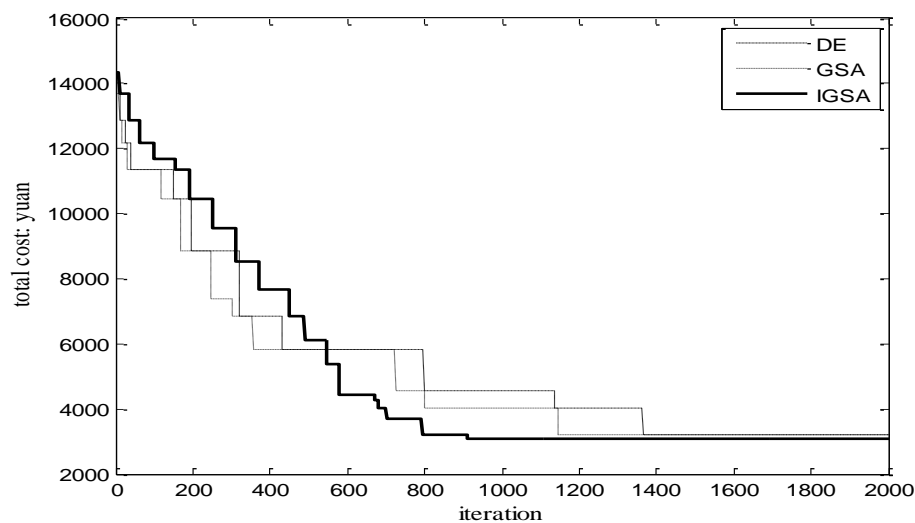


Figure 1. The Performance Comparisons between DE, GSA and IGSA

It is not hard to see that the convergence speed of differential evolution algorithm and basic GSA are fast at the beginning of calculation, but they fall into local optimum afterwards, and fail in getting the global optimal solution. For IGSA, while the initial search speed of algorithm is slow, but the algorithm can continue to optimize in the mid and later stage, which can calculate the optimal solution of multi-depot and multi-vehicle emergency vehicle scheduling problem more accurately, thereby providing a new way to solve this kind of problems.

7. Conclusions

This paper establishes a mathematical model of multi-depot and multi-vehicle emergency vehicle scheduling problem, and put forward the IGSA to solve this problem. The algorithm improves gravity coefficient and velocity selection formula, and selects updated particle's location according to the survival of the fittest. Simulation results show that the improved GSA is feasible and effective in solving this kind of problems.

Acknowledgments

This work was supported by the National Social Science foundation of China (No. 15CGL001), the Science and Technology Support Project of Gansu Province (No. 1304FKCA097) and the Basic Scientific Research Expenses of Finance Department of Gansu Province (No. 214150).

References

- [1] L. Wang, "Multi-depot emergency scheduling research", *Journal of railway transportation and economy*, vol. 32, no. 7, (2010), pp. 47-52.
- [2] Y. Gan, S. Lv and J. Li, "Consider cost more to save some more supplies emergency scheduling research", *Chinese journal of safety science*, vol. 21, no. 9, (2011), pp. 172-176.
- [3] W. Fan, "Intergrated optimization of depot selection and vehicle routing for relief resource distribution in urban public emergencies", *Chengdu: SouthWest JiaoTong University*, no. 05, (2011).
- [4] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "BGSA: Binary gravitational search algorithm", *Natural Computing*, vol. 9, no. 3, (2010), pp. 727-745.
- [5] P. Li, "Improved gravitational search algorithm and applied in flatness pattern recognition", *Qin huangdao: Yanshan University*, no. 05, (2013).
- [6] R. Esmat, H. Nezamabadi-pour and S. Saryazdi, "GSA: A gravitational search algorithm", *Inform Sci*, (2009), no. 179, pp. 2232-2248.
- [7] S. Sarafrazi, H. Nezamabadi-pour and S. Saryazdi, "Disruption: A new operator in gravitational search algorithm", *Sci Iran*, no. 18, (2011), pp. 539-548.
- [8] B. Schutz, "Gravity from the ground up", London: Cambridge University Press, (2003).
- [9] P. Li and H.B. Duan, "Path planning of unmanned aerial vehicle based on improved gravitational search algorithm", *Sci China Tech Sci*, no. 55, (2012), pp. 2712-2719.
- [10] M. Eslami, H. Shareef, A. Mohamed and M. Khajezadeh, "Gravitational search algorithm for coordinated design of PSS and TCSC as damping controller", *Journal of Central South University*, no. 19, no. 4, (2012), pp. 923-932.
- [11] H. Askari and S.-H. Zahiri, "Decision function estimation using intelligent gravitational search algorithm", *International Journal of Machine Learning and Cybernetics*, vol. 3, no. 2, (2012), pp. 163-172.
- [12] M. Khajezadeh, M. Raihan Taha A. El-Shafie and M. Eslami, "A modified gravitational search algorithm for slope stability analysis", *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, (2012), pp. 1589-1597.

