

# Comparison of optimization methods based on GSA

Nihan Kazak<sup>1</sup>, Nesibe Yalçın<sup>1</sup>, Ali Erdem Çerçevik<sup>2</sup>

Accepted : 12/05/2017 Published: 21/08/2017

**Abstract:** In recent years, many heuristic evolutionary optimization algorithms have been developed. Gravitational search algorithm (GSA) is one of these algorithms. It is inspired by Newton's law of universal gravitation. In this paper, we compare two modified algorithms based on GSA. One of the algorithms is called Member-Satellite algorithm. Members are randomly positioned in the search space and a certain amount of satellites are assigned around the members in the predetermined region. Members and their satellites are used to find a near optimal solution all together. The second one is MSS-GSA. When interconnected objects are used, it is possible to obtain a solution closer to the optimum point. For this reason mass spring system is integrated into the GSA. Three benchmark functions are used to compare performance. Experimental results show that the highest performance is obtained with Member-Satellite algorithm.

**Keywords:** Gravitational Search Algorithm, Heuristic Algorithm, Optimization, Swarm Intelligence

## 1. Introduction

Optimization techniques are used to solve many engineering problems. Recently, in addition to the classical optimization methods, heuristic optimization methods have been developed to find the best, the optimum result under the given conditions. Gravitational search algorithm (GSA) is one of the heuristic optimization methods [1]. The algorithm is intended to improve the performance in the exploration and exploitation capabilities of a population based algorithm, based on gravity rules [2]. GSA has been verified high quality performance in solving different optimization problems in the literature [3],[4],[5],[6] and [7] and there are many improved version of GSA [8], [9],[10] and [11].

In [12], we proposed Member-Satellite Algorithm (MSA) in order to improve the searching ability of GSA and to offer faster convergence while maintaining and improving quality of the solution. In the proposed MSA, after agents are positioned in search space, new agents known as satellites are positioned around the agents. The search is then performed by the agents and by the satellites as well. Satellites, except for the calculation of the fitness values, are not included in the other computational processes. So, computational load of algorithm is reduced. Therefore, the algorithm realizes more detailed searching in the related space than standard GSA.

In [13], we proposed MSS-GSA which is based on Mass Spring System principle. Mass spring system which is used frequently in deformation of the objects is integrated into the GSA. In the optimization methods, the objects assigned to the search space search the optimum solution independently of each other. When using interconnected objects, it is possible to obtain a closer solution to the optimum point. For this reason, springs are linked between masses like MSS. The masses are pulled towards the best solution by applying force to each other in obtained MSS system. The algorithm scans a wider area by both global and local search in search space.

The outline of the paper is as follows: The brief introduction to GSA is given in Section II, MSS-GSA and MSA are introduced in Section III and Section IV, respectively. In Section V the

experimental results are demonstrated and finally in Section VI the conclusions are given in the last section.

## 2. Brief Review of GSA

GSA is an optimization method based on Newton's acceleration law, the second of motion laws, and Newton's universal gravitation law. Each mass in search space attract each others with gravitational force. A more effective mass which is the heaviest mass moves slower than the others and attracts the others better. At the end of the number of iterations or any termination is reached, the object with the greatest mass will be the optimal solution of the problem.

It is assumed that there are  $N$  masses, the position of the  $i$ th agent is as follows:

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

Gravitational constant is defined as:

$$G(t) = G_0 \times e^{(-\alpha \frac{t}{T})} \quad (2)$$

The current fitness value of each mass is calculated by fitness function. The best and the worst of the calculated fitness values are selected.

According to [1], each mass is calculated as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (3)$$

With

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (4)$$

The heaviest mass is the most effective. The force acting on mass  $i$  from mass  $j$  in the  $d$ th dimension at a specific time  $t$  as following:

$$F_{ij}^d(t) = G(t) \times \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \epsilon} \times (X_j^d(t) - X_i^d(t)) \quad (5)$$

Where  $R_{ij}(t)$  represents Euclidean distance between  $i$ th mass and  $j$ th mass.  $\epsilon$  is a small constant.

Total force that acts on agent  $i$  in a dimension  $d$  is computed as follows:

$$F_i^d(t) = \sum_{j \in K_{best, j \neq i}}^N rand_j F_{ij}^d(t) \quad (6)$$

<sup>1</sup> Department of Computer Engineering, Bilecik Seyh Edebali University Bilecik/Turkey

<sup>2</sup> Department of Civil Engineering, Bilecik Seyh Edebali University Bilecik/Turkey

\* Corresponding Author: Email: nihan.kazak@bilecik.edu.tr

Kbest is reduced linearly with time from the beginning  $K_0$  until one. So only the best solution at the end of the algorithm attracts other objects in the search space. Other objects can not apply a force to the best solution. The law of motion is used for the calculation of agent acceleration:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (7)$$

The velocity and position updating equations:

$$V_i^d(t+1) = \text{rand}_i \times V_i^d(t) + a_i^d(t) \quad (8)$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \quad (9)$$

If a termination criterion is met, the algorithm stops.

### 3. MSS-GSA

The aim of MSS-GSA is to search of the optimum solution by considering the principle of the mass spring system. For this reason, a spring is connected between the masses used in the GSA. Total force applied to mass is calculated by addition of external forces that arising from the potential difference due to the location of the masses and internal forces generated by the spring which is connected to the mass. It is aimed to the global and local research in search space by this way. The motion of masses in search space is based on Newton's acceleration law.

In the algorithm, the boundaries of the search space are determined first. Masses are located randomly in created search space.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^N) \quad i = 1, 2, \dots, N \quad (10)$$

where  $x_i^d$  is the location at  $d$ th dimension of  $i$ th mass. After the members are defined, they are sent to the fitness function and the fitness values of each are calculated. The best value of fitness values is decided by taking care whether the problem is minimization or maximization problem:

$$\text{best}(t) = \min_{j=\{1, \dots, N\}} \text{fit}_j(t) \quad (11)$$

$$\text{worst}(t) = \max_{j=\{1, \dots, N\}} \text{fit}_j(t) \quad (12)$$

where  $\text{fit}_j(t)$  represents the fitness of the  $j$ th mass at iteration  $t$ ,  $\text{best}(t)$  and  $\text{worst}(t)$  represent the best and worst fitness value of all agents at generation  $t$ .

When searching for the optimum solution, the search speed needs to be checked. The gravitational constant and the spring constant are reduced to provide spring stiffness and search speed control:

$$G(t) = G_0 \times e^{-\frac{\alpha t}{T}} \quad (13)$$

$$K(t) = k_0 \times e^{-\frac{\alpha t}{T}} \quad (14)$$

where  $G(t)$  with initial value  $G_0$  is named as the gravitational constant and  $K(t)$  is the spring stiffness at time  $t$ .  $T$  and  $t$  represent total iteration number and current iteration number, respectively. In a minimization problem with global and local minimums, as in Fig. 1, members are closer or farther away from the global minimum relative to their fitness values when they are randomly distributed in the search space.

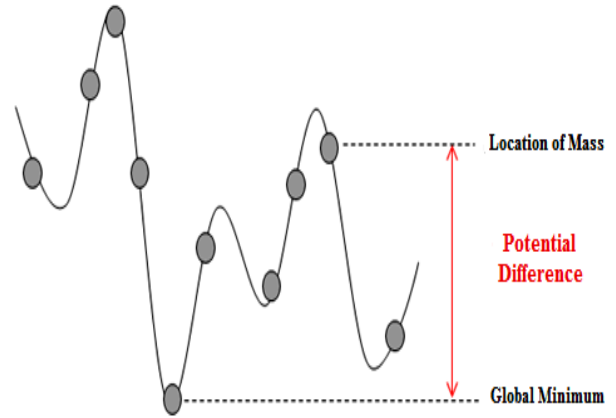


Fig. 1. Potential difference between masses

The mass with the lowest fitness value is the closest mass to the global minimum. In Fig. 1, it is clearly seen that the distance between the masses causes a potential difference. An external force occurs because of this potential difference. In other word, the external force arise due to potential difference of mass location that variation of fitness values of masses in search space.

$$F_{\text{ext}_{ij}}^d(t) = G(t) \times \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \times \left( \frac{X_j^d(t) - X_i^d(t)}{|X_j^d(t) - X_i^d(t)| + \epsilon} \right) \quad (15)$$

$$F_{\text{ext}_i^d}(t) = \sum_{j \neq i}^N \text{rand}_j F_{\text{ext}_{ij}}^d(t) \quad (16)$$

In Eq ..., the external force on  $i$ th mass by  $j$ th mass is calculated.  $|X_j^d(t) - X_i^d(t)|$  represents distance between  $i$ th and  $j$ th mass in a dimension  $d$ .  $(X_j^d(t) - X_i^d(t))$  represents difference location of  $i$ th and  $j$ th mass in a dimension  $d$ .  $\epsilon$  is a small constant value. Fitness values calculated masses is sorted best to worst. A mass spring system as in Fig. 2 is formed by linking spring between the ordered masses.

Each mass apply force to masses which are linked with spring. (17) is used for zeroes to potential difference.

$$F_{\text{int}_{ij}}^d(t) = K \times \frac{\text{fit}_i(t) - \text{fit}_j(t)}{\text{best}(t) - \text{worst}(t)} \times \left( \frac{X_i^d(t) - X_j^d(t)}{|X_i^d(t) - X_j^d(t)| + \epsilon} \right) \quad (17)$$

Springs apply a same quantity and reverse direction force to force which draws or stresses to itself:

$$F_{\text{int}_{ji}}^d(t) = -F_{\text{int}_{ij}}^d(t) \quad (18)$$

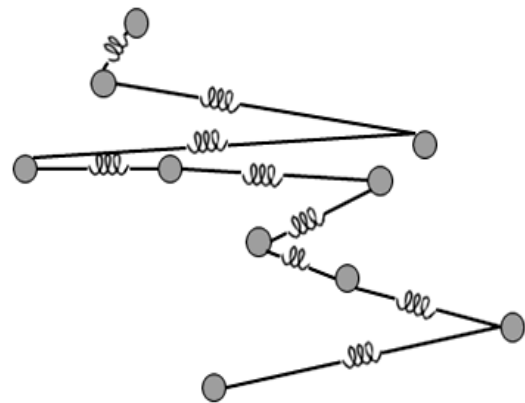


Fig. 2. Linking springs

In (18),  $j$ th mass is the next of  $i$ th mass linked with spring.  $F_{\text{int}_{ji}}^d(t)$  represents internal force on  $j$ th mass by  $i$ th mass. According to the above equations, masses are drawn to the best

mass. In this case, local search in the search space is also performed.

Finally, the internal and external forces applied by one mass to the other masses are summed and the total force is calculated:

$$F_i^d(t) = F_{int_i}^d(t) + F_{ext_i}^d(t) \quad (19)$$

where  $F_i^d(t)$  represents total force on  $i$ th mass in dimension  $d$ . The rest of the algorithm, calculation of speed and displacement, is proceeds as in the GSA. If no stopping criterion is found, the algorithm is continued by returning to the step where the fitness values of the masses are calculated.

#### 4. Member – Satellite Algorithm

GSA method is modified to make more surface scan in the proposed method, called Member-Satellite Algorithm (MSA). Each mass in the search space is named as a member; masses located around the members are named as satellite in MSA. This structure is demonstrated in Fig. 3. Large circles coloured differently with black frame represents members and small circles with same colour represent their satellite. The satellites are placed randomly to related members within a certain radius.

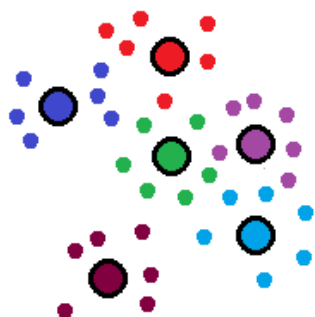


Fig. 3. Member – Satellite System

Fitness values of members and their satellites are computed according to fitness function. A satellite having the best fitness value labelled as a member in case of its fitness value is better. Optimum solution is searched together with members and their satellites. Satellites, except for the calculation of the fitness values, are not included in the other computational processes. So, computational load of algorithm is reduced. In the algorithm, the boundaries of the search space are determined first. Members are located randomly in search space:

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad i = 1, 2, \dots, N \quad (20)$$

A fixed number of satellites are assigned for each member.

$$X_{is} = (x_{is}^1, \dots, x_{is}^d, \dots, x_{is}^n) \quad i = 1, 2, \dots, N \quad s = 1, 2, \dots, M \quad (21)$$

where  $x_{is}^d$  represents location of  $s$ th satellite of  $i$ th member in  $d$  dimension.

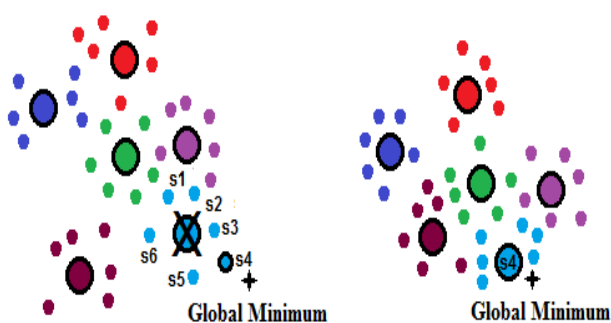


Fig. 4. From satellite to member

Fitness values of members and their satellite are computed. If fitness value of a member is better than any its satellite, member label continue and position of this is protected. Otherwise, the satellite having better fitness value becomes the member. The best and the worst member are determined in all members with obtained members and proposed algorithm is continued using GSA processes.

There is a pseudo-code of MSA:

```

FOR
  Assign N members randomly in search space.
   $X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad i = 1, 2, \dots, N$ 
END
DO FOR
  Assign M satellites randomly around of each members in a
  certain region
   $X_{is} = (x_{is}^1, \dots, x_{is}^d, \dots, x_{is}^n) \quad i = 1, 2, \dots, N \quad s = 1, 2, \dots, M$ 
  Compute Fitness Values
  IF Satellite of Member > Member
    Member = Satellite
  END
  Update Gravity Constant, Best and Worst
  Compute Mass of Members
  Compute Force on Each Member
  Compute Acceleration
  Update Speed and Location
END
WHILE Termination Criteria
  
```

#### 5. Experimental Results

In this section, GSA, MSS-GSA and MSA are evaluated on 3 benchmark functions. Algorithms were implemented in MATLAB language and run on a PC with an Intel Core i5-CPU M 430 @ 2.27 GHz and 4 GB memory.

Sphere function, which is the first benchmark function, has a single global minimum. This type of functions is called single-mode functions. Sphere functions also have a feature of convex and smooth.

$$F_{\text{Sphere}}(X) = \sum_{i=1}^n x_i^2 \quad (22)$$

where,  $n$  represents dimensions of function. The global minimum is 0 for this function. Its admissible range is  $[-100 \ 100]^n$  and its dimension is 30.

The second benchmark function is Ackley function. It has several local optimum points but one global optimum. This type of functions is called multi-mode functions.

$$F_{\text{Ackley}}(X) = -20 \times \exp\left(-0.2 \times \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}\right) - \exp\left(\frac{\sum_{i=1}^n \cos(2\pi x_i)}{n}\right) + 20 + e \quad (23)$$

where  $n$  represents dimensions of function. The global minimum is 0 for this function. Its range is  $[-5 \ 5]^4$  and its dimension is 2.

The third function is humpcamel function which is one of the multi-modal functions. It has 6 local optimum together with 2 global optimum.

$$F_{\text{Humpcamel}}(X) = 4 \times x_1^2 - 2.1 \times x_1^4 + \frac{x_1^6}{3} + x_1 \times x_2 - 4 \times x_2^2 + 4 \times x_2^4 \quad (24)$$

Its admissible range is  $[-32 \ 32]^n$  and its dimension is 30. All the functions are to be minimized. In all cases for both algorithms, population size is set to 25,  $G_0$  and  $\alpha$  are considered equal to 50 and 10, respectively.  $K_0$  is set to 50 and is decreased exponentially to 1.

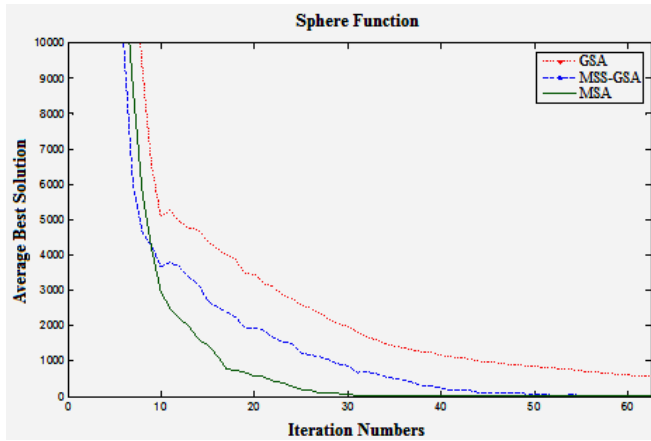


Fig. 5. Performance of the algorithms for Sphere Function

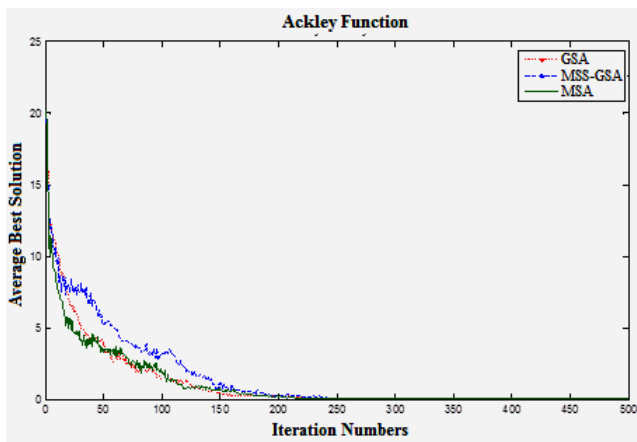


Fig. 6. Performance of the algorithms for Ackley Function

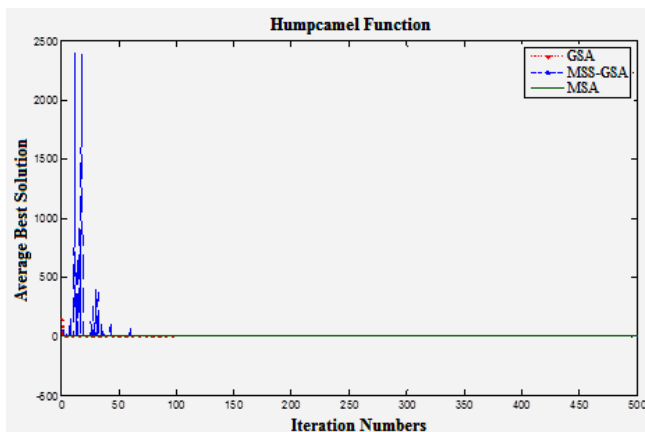


Fig. 7. Performance of the algorithms for Humpcamel Function

## 6. Conclusions

In this study we compare Mass-Spring System-Gravitational Search Algorithm (MSS-GSA) and Member-Satellite Algorithm (MSA) based on GSA. In MSS-GSA, masses in search space are linked with springs so mass-spring system is applied. In MSA, each mass called member have own satellites and optimum solution is search together. In this way, searching area is enlarged. The algorithms are tested on three benchmark functions.

Comparative experimental results demonstrate that MSA gets better result than the other algorithms. When algorithms are compared in terms of computation time, GSA converges to optimum value in a shorter time. In the future, we will make further study on how to apply the MSA into real world optimization problems.

## References

- [1] E. Rashedi, H. Nezamabadi, and S. Saryazdi, "GSA: A Gravitational Search Algorithm", *Information Sciences*, 179(13): 2232–2248 (2009).
- [2] N. M. Sabri, P. Mazidah, and R. M. Mohamad, "A review of gravitational search algorithm." *Int. J. Advance. Soft Comput. Appl* 5.3 (2013): 1-39.
- [3] S. Zhang, C. Li, J. Zhou, "Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm", *Energy Convers Manage*, 52 (2011), pp. 374–381.
- [4] B. Zibanezhad, K. Zamanifar, N. Nematbakhsh, and F. Mardukhi, "An approach for web services composition based on QoS and gravitational search algorithm." *In Proceedings of the innovations in information technology conference*, 2010. p 340–344.
- [5] H. R. Hassanzadeh, and M. Rouhani, "A multi-objective gravitational search algorithm." *In: Proceedings of the communication systems and network conference*, 2010. p. 7–12.
- [6] S.R. Balachandar, K. Kannan. "A meta-heuristic algorithm for set covering problem based on gravity", *Int J Comput Math Sci*, 4 (2010), pp. 223–228.
- [7] S. Duman, U. Güvenç, N. Yörükeren, "Gravitational search algorithm for economic dispatch with valve-point effects", *Int Rev Electr Eng*, 5 (6) (2010), pp. 2890–2895.
- [8] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "BGSA: Binary gravitational search algorithm", *Nat. Comput.* 9 (2010) 727–745.
- [9] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl and M.-B. Radac, "Novel adaptive gravitational search algorithm for fuzzy controlled servo systems", *IEEE Trans. Ind. Inf.* 8(4) (2012) 791–800
- [10] M. B. Dowlatshahi, H. Nezamabadi-pour and M. Mashinchi, "A discrete gravitational search algorithm for solving combinatorial optimization problems", *Inf. Sci.* 258 (2014) 94–107.
- [11] X.-H. Han, X.-M. Chang, L. Quan, X.-Y. Xiong, J.-X. Li, Z.-X. Zhang and Y. Liu, "Feature subset selection by gravitational search algorithm optimization", *Inf. Sci.* 281 (2014) 128–146.
- [12] Kazak N. and Duysak A., "Modified Gravitational Search Algorithm.", *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2012, pp.1-4
- [13] Kazak N. and Duysak A., "Geliştirilmiş Yerçekimsel Arama Algoritması: MSS-GSA", *3<sup>rd</sup>. International Symposium on Innovative Technologies in Engineering and Science (ISITES)*, 2015.