Exploitative Gravitational Search Algorithm

Aditi Gupta(✉), Nirmala Sharma, and Harish Sharma

Rajasthan Technical University, Kota, India
aditi.2404.gupta@gmail.com, harish.sharma0107@gmail.com

Abstract. Gravitational search algorithm (GSA) is a simple well known meta-heuristic search algorithm based on the law of gravity and the law of motion. In this article, a new variant of GSA is introduced, namely Exploitative Gravitational Search Algorithm (EGSA). In the proposed EGSA, two control parameters (Kbest and Gravitational constant) are modified that play an important role in GSA. Gravitation constant G is reduced iteratively to maintain a proper balance between exploration and exploitation of the search space. Further, To enhance the searching speed of algorithm *Kbest* (best individuals) is exponentially decreased. The performance of proposed algorithm is measured in term of reliability, robustness and accuracy through various statistical analyses over 12 complex test problems. To show the competitiveness of the proposed strategy, the reported results are compared with the results of GSA, Fitness Based Gravitational Search Algorithm (FBGSA) and Biogeography Based Optimization (BBO) algorithms.

Keywords: Gravitational search algorithm · Swarm intelligence · Heuristic search algorithm · Elitism · Exponential · Gravitational constant

1 Introduction

Nature is an origin of inspiration for solving hard and complex problems. Nature-inspired algorithms (NIAs) are inspired by nature and used to deal with difficult real-world engineering problems [6]. Swarm intelligence algorithms [1,2] are inspired by any type of collective behaviours of individuals in nature. Gravitational search algorithm (GSA) [6] is a swarm intelligence type algorithm that is inspired by the Newton's physics concept gravitational force and motion of individuals in nature. Individuals fascinate each others by the gravity force and accelerate according to the force applied on individuals. Individuals with heavier masses have high attraction power compare to lower masses individuals. By this attraction power of individuals higher masses individuals move slowly as compare to lower masses individuals. GSA is an optimization algorithm and provides proper balancing between exploitation and exploration capabilities. So in this algorithm, heavier masses individuals are responsible for exploitation whereas lighter masses individuals are responsible for the exploration of the search area.

K. Deep et al. (eds.), *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 546, DOI 10.1007/978-981-10-3322-3_15

When searching process start lighter masses (individuals are far from the optimum solutions) individuals move with large step size (exploration) and after this when individuals converge to the optimum solutions i.e. higher masses individuals move with comparative small step size (exploitation). Researchers have been progressively produce new techniques to refine the performance of the algorithm [3,7,8,11].

In this paper, to improve the searching capability and diversification and intensification proficiency of GSA algorithm, a new variant of GSA is designed, namely Exploitative Gravitational Search Algorithm (EGSA). In the proposed EGSA, $Kbest$ (best individuals) and Gravitational constant G are modified such that searching efficiency of GSA is increased and the solution explore the search space in early iteration having large step size while exploit the identified search region in later iterations with small step size. To balance the number of individuals that applied force to other individuals as the iteration is increased, Kbest value is exponentially decreased. Gravitational constant is also decreased through iterations that is responsible for the step size of individuals.

The remaining paper is organised as shown: In Sect. 2, a brief overview of GSA is illustrated. Exploitative GSA algorithm (EGSA) is proposed in Sect. 3. In Sect. 4, performance of EGSA is tested with several numerical benchmark functions. Finally, Sect. 5 gives a summary and conclude the work.

2 Gravitational Search Algorithm

E. Rashedi et al. developed Gravitational Search Algorithm (GSA) in 2009 [6]. GSA is a population-based stochastic search algorithm inspired by the Newton's gravitational law and movement of individuals in universe by gravitational force. According to Newton's gravity law "*Every individuals in-universe fascinate each other with force, this force is directly proportional to the product of individuals masses and inversely proportional to the square of the distance between individuals masses* [6]". Force applied to the individuals, by this force individuals are accelerated from their position. Performance of individuals is measure by their mass. Agents with higher mass are good as compare to lighter mass agents.

The GSA algorithm is described as follows: Each individual X_i in search space with I number of individuals is represented as:

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n) \qquad for\ i = 1, 2, ..., I, \tag{1}$$

here x_i^d shows the position of i^{th} individual in d dimensional area.

Mass of individuals is a based on the individuals fitness. The fitness of all individuals are calculated and worst and best fitness are identified for calculating the mass of individuals.

- For minimization problems best and worst fitness are:

$$best(g) = minfit_j(g) \qquad j \in 1, \cdot\cdot\cdot, I \tag{2}$$

$$worst(g) = maxfit_j(g) \qquad j \in 1, \cdot\cdot\cdot, I \tag{3}$$

- For maximization problems best and worst fitness are:

$$best(g) = maxfit_j(g) \qquad j \in 1, \cdots, I \tag{4}$$

$$worst(g) = minfit_j(g) \qquad j \in 1, \cdots, I \tag{5}$$

  $maxfit_j(g)$ and $minfit_j(g)$ show the maximum and minimum fitness value of the j^{th} individual at iteration g.

In GSA inertia, active and passive gravitational masses are equal. Individual with heavier masses are more efficient. Heavier masses individuals have higher attraction power and move slowly. Masses in GSA depend on the fitness value of individuals and calculated as follows:

$$M_{aj} = M_{pi} = M_{ii} = M_i, \qquad i = l, 2, ..., I. \tag{6}$$

$$m_i(g) = \frac{fit_i - worst(g)}{best(g) - worst(g)} \tag{7}$$

$$M_i = \frac{m_i(g)}{\sum_{j=1}^{I} m_j(g)} \tag{8}$$

here M_{ii} and M_{pi} are inertia and passive gravitational masses of i^{th} individual respectively and M_{aj} is active gravitational mass of j^{th} individual. fit_i is the fitness value of i^{th} individual.
$G(g)$ is the gravitational constant computed as Eq. 9.

$$G(g) = G_0 e^{(-\alpha g/MaxIt)} \tag{9}$$

Here, G_0 and α are constant and initialized at the starting. The value of $G(g)$ is reduced exponentially during each iteration for controlling search accuracy. $MaxIt$ is total number of iteration. Acceleration of individuals depends upon the ratio of force and mass of the individual [5] and calculated as follows:

$$a_i^d(g) = F_i^d(g)/M_{ii}(g) \tag{10}$$

$F_i^d(g)$ is the overall force acting on i^{th} individual computed as:

$$F_i^d(g) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(g) \tag{11}$$

$Kbest$ is computed as follows:

$$Kbest = finalper + (1 - \frac{g}{MaxIt}) \times (N - finalper) \tag{12}$$

$$Kbest = round(N \times \frac{Kbest}{N}) \tag{13}$$

Here, $finalper$ is the constant and N is the total number of individuals in the search space. $Kbest$ is initial N individuals with the best fitness value and highest mass. Kbest will reduce linearly in each iteration and at the final only one individual applying force to the other individuals.

Force on i^{th} individual by j^{th} individuals mass during iteration g is computed using the following Eq. 14:

$$F_{ij}^{d}(g) = G(g).(M_{pi}(g) \times M_{aj}(g)/R_{ij}(g) + \varepsilon).(x_j^d(g) - x_i^d(g)) \tag{14}$$

Here, $R_{ij}(g)$ is the Euclidian-distance between two individuals i and j at iteration g. Gravitational constant $G(g)$ is calculated using Eq. 9 while ε is a small constant. The velocity update equation for individuals is defined as:

$$v_i^d(g+1) = rand_i \times v_i^d(g) + a_i^d(g) \tag{15}$$

here, $rand$ is random variable in interval [0, 1]. $v_i^d(g)$ and $v_i^d(g+1)$ are the velocity of i^{th} individual at the iteration g and $g+1$ subsequently.

The position update equation for individuals is defined as:

$$x_i^d(g+1) = x_i^d(g) + v_i^d(g+1) \tag{16}$$

here, $x_i^d(g)$ and $x_i^d(g+1)$ are the position of i^{th} individual at the iteration g and $g+1$ subsequently. Velocity of individuals is updated during each iteration. Due to changes in the velocity every individual update its position.

This procedure is carry on until their termination criteria is met or iteration reach their maximum limit.

3 Exploitative Gravitational Search Algorithm

In the population-based algorithms, behavior of agent is measured by the exploitation and exploration capability in the search space. Exploration is for the enlarging the entire search space and exploitation is the finding optimum solution from the previously visited good solutions. During the early iteration of the algorithm, GSA visit the entire search space to find out the optimal solutions. After the lapse of iteration GSA exploit the search space by visiting the previously visited points. For the better performance of any population-based algorithm it is necessary to maintain a proper balance between the exploitation and exploration. Initially, when the individuals are not converged, exploration is needed to find out the good solutions in the whole search space. For the exploration large step size is necessary. After the lapse of iteration, individuals are converged. Hence for finding the optimal solution of the algorithm, individuals needs to exploit the search space (step size is comparatively less). In GSA gravitational constant G affects the step size of individuals. As mentioned in Eq. 14, force is directly proportional to the gravitational constant G and from the Eq. 10, acceleration is depend on the force of individuals. Acceleration in GSA plays a vital role for the step size of the individuals. Therefore in this paper gravitational constant G is modified as shown here:

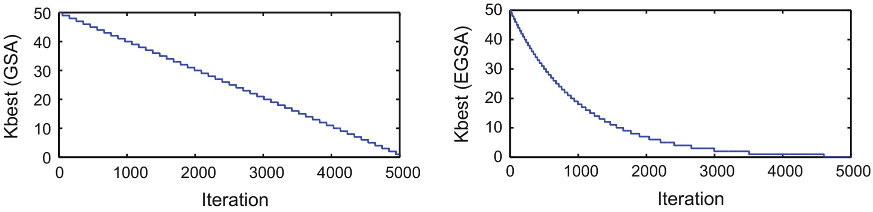
$$G(g) = G_0 e^{(-\alpha g/MaxIt)}(1 - \frac{g}{MaxIt}) \tag{17}$$

From the Eq. 17, it is clear that the value of gravitational constant will be high during the initial iteration and value is reduced iteratively. Therefore, the acceleration and step size of the individuals are decreased as the number of iteration increased.

$Kbest$ in Eq. 13 controls the number of individuals that apply the force to other individuals in search space. A large number of $Kbest$ (individuals) means large number of individuals interact with each other and movement between the individuals is high. As the result convergence speed is lower. From Eq. 13 it is clear that $Kbest$ is linearly decreased therefore change in $Kbest$ is very small as the number of iterations increase. Due to this movement and interaction between individuals is comparatively lower but the effect in convergence speed is not much. Therefore in this paper $Kbest$ is modified as shown here:

$$Kbest = round(N \times exp(-\beta g/MaxIt)) \tag{18}$$

Here, N is total number of individuals and β is constant. The value of $Kbest$ is reduced exponentially during each iteration. From the Eq. 18 it is clear that $Kbest$ is exponentially decreased. At initial iteration $Kbest$ is large therefore movement and interaction between the individuals is more that show the exploration of the search space. Whereas the number of iteration is increased $Kbest$ is reduced therefore movement and interaction between the individuals is comparative less that shows the exploitation of the individuals.



(a) Linearly decrement in $Kbest$ (GSA)

(b) Exponentially decrement in $Kbest$ (EGSA)

Fig. 1. Effect of $Kbest$ in number of individuals

Behaviour of $Kbest$ through iterations is shown in Fig. 1. It is clear from this figure that in EGSA, $Kbest$ is decreases exponentially through iterations whereas in GSA $Kbest$ is decreases linearly through iteration. So initially large number of individuals apply force as the number of iteration is increased comparatively less number of individuals apply force. So, EGSA regulates a proper balance between diversification and intensification proficiency and improve the searching ability as the number of iteration increase. The pseudo-code of EGSA is shown in Algorithm 1.

```
– Identification of search area.
– Creating a randomly dispersed set of individuals;
while Stopping condition is not satisfied do
    – Calculate fitness of individuals.
    – Calculate individuals mass by Eq. 6 and 7.
    – Evaluate constant ($G$) by Eq. 17.
    – Evaluate $Kbest$ by Eq. 18.
    – Calculate force ($F$) for each direction by Eq. 14 and acceleration ($a$) for
      every individuals by Eq. 10.
    – Update individuals velocity by Eq. 17 and position by Eq. 18.
end
```

Algorithm 1. EGSA Algorithm

4 Results and Discussions

To examine the outcome of the EGSA, 12 different benchmark functions (f_1 to f_{12}) are picked as shown in Table 1.

To certify the prosection of the proposed algorithm EGSA, a comparative analysis is carried out among EGSA, standard GSA, FBGSA [4] and BBO [10]. To authenticate the performance of the considered algorithms over the test problems, the experimental setting is given below:

- The number of simulations/run = 30,
- Number of population (N) = 50,
- $G_0 = 100$, $\alpha = 20$, $\beta = 5$ and $finalper = 2$,
- Experimental settings for the algorithms GSA, FBGSA [4] and BBO [10] are simulated from their primary research papers.

Table 2 display the experimental results of the examine algorithms. A detailed analysis about the standard deviation (SD), mean error (ME), average number of function evaluations (AFE) along with the success rate (SR) are shown in Table 2. Results in Table 2 replicates, many times EGSA exceeds in terms of reliability, efficiency as well as accuracy as compare to the GSA, FBGSA and BBO.

Further, Mann-Whitney U rank sum test [9] is performed at 5% level of remarkable ($\alpha = 0.05$) between EGSA - GSA, EGSA - FBGSA and EGSA - BBO. Table 3 display the compared results of mean function evaluation and Mann-Whitney test for 30 simulations. In Mann-Whitney test, we observe the remarkable difference between two data set. If remarkable difference is not seen then = symbol appears and when remarkable difference is observed then comparison is performed in terms of the AFEs. And we use + and - symbol, + represent the EGSA is superior than the examined algorithms and - represent the algorithm is inferior. The last row in Table 3, authorize the excellence of EGSA over GSA, FBGSA and BBO.

Table 1. Test problems, D: Dimension, AE: Acceptable Error

Test problem	Objective function	Search range	Optimum value	D	AE
Alpine	$f_1(x) = \sum_{i=1}^{D} \lvert x_i \sin\ x_i + 0.1x_i \rvert$	$[-10\ 10]$	$f(0) = 0$	30	$1.0E-05$
Zakharov	$f_2(x) = \sum_{i=1}^{D} x_i{}^2 + (\sum_{i=1}^{D} \frac{ix_i}{2})^2 + (\sum_{i=1}^{D} \frac{ix_1}{2})^4$	$[-5.12\ 5.12]$	$f(0) = 0$	30	$1.0E-02$
Schwefel	$f_3(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10\ 10]$	$f(0) = 0$	30	$1.0E-05$
Step function	$f_4(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100\ 100]$	$f(-0.5 \leq x \leq 0.5) = 0$	30	$1.0E-05$
Neumaier 3 problem (NF3)	$f_5(x) = \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$	$[-900\ 900]$	$f(0 = -(D*(D+4)*(D-1))/6.0)$	10	$1.0E-05$
Colville	$f_6(x) = 100[x_2 - x_1^2]^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 + 10.1[(x_2-1)^2 + (x_4-1)^2] + 19.8(x_2-1)(x_4-1)$	$[-10\ 10]$	$f(1) = 0$	4	$1.0E-05$
Gear train problem	$f_7(x) = ((1/6.931) - ((x(1)*x(2))/(x(3)*x(4))))^2$	$[-4\ 4]$	$f(1) = 4$	4	$1.0E-05$
Six-hump camel back	$f_8(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4+4x_2^2)x_2^2$	$[-5\ 5]$	$f(-0.0898, 0.7126) = -1.0316$	2	$1.0E-05$
Easom's function	$f_9(x) = -\cos x_1 \cos x_2 e^{((-(x_1-\pi)^2-(x_2-\pi)^2))}$	$[-100\ 100]$	$f(\pi, \pi) = -1$	2	$1.0E-13$
Hosaki problem	$f_{10(x)} = (1 - 8x_1 + 7x_1^2 - 7/3x_1^3 + 1/4x_1^4)x_2^2 \exp(-x_2)$, subject to $0 \leq x_1 \leq 5, 0 \leq x_2 \leq 6$	$[0\ 5]$, $[0\ 6]$	-2.3458	2	$1.0E-05$
McCormick	$f_{11}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	$-1.5 \leq x_1 \leq 4, -3 \leq x_2 \leq 3$	$f(-0.547, -1.547) = -1.9133$	30	$1.0E-04$
Moved axis parallel hyper-ellipsoid	$f_{12}(x) = \sum_{i=1}^{D} 5i \times x_i^2$	$[-5.12\ 5.12]$	$f(x) = 0; x(i) = 5 \times i, i = 1 : D$	30	$1.0E-12$

Table 2. Comparison of the results of test functions, TP: Test Problem

TP	Algorithm	SD	ME	AFE	SR
f_1	EGSA	4.76E-07	9.44E-06	127540.00	30
	GSA	5.59E-07	9.30E-06	154615.00	30
	FBGSA	5.57E-07	9.25E-06	141046.67	30
	BBO	5.78E-03	1.04E-02	200000.00	0
f_2	EGSA	1.01E-04	9.89E-03	84075.00	30
	GSA	1.98E+00	3.61E+00	200000.00	0
	FBGSA	8.65E+00	4.04E+01	200000.00	0
	BBO	5.28E-03	1.77E-02	200000.00	0
f_3	EGSA	4.65E-07	9.50E-06	151536.67	30
	GSA	4.94E-07	9.42E-06	181821.67	30
	FBGSA	6.47E-07	9.23E-06	166096.67	30
	BBO	5.61E-02	5.61E-02	20000.00	0
f_4	EGSA	1.25E-04	1.54E-01	8996.67	30
	GSA	4.96E-03	2.56E+02	11583.33	30
	FBGSA	5.60E-04	4.15E-02	12635.00	30
	BBO	4.40E+01	4.80E-05	5351.67	30
f_5	EGSA	1.49E-06	8.45E-06	67220.00	30
	GSA	1.55E-06	7.73E-06	86376.67	30
	FBGSA	1.86E-06	7.45E-06	81458.33	30
	BBO	5.49E-02	1.42E-01	200000.00	0
f_6	EGSA	1.60E-04	9.02E-04	74060.00	30
	GSA	5.23E-02	1.88E-02	142220.00	26
	FBGSA	3.73E-01	9.92E-02	200000.00	0
	BBO	1.41E+00	2.73E-01	191210.00	3
f_7	EGSA	7.92E-13	1.83E-12	14473.33	30
	GSA	8.88E-13	1.85E-12	21290.00	30
	FBGSA	7.81E-13	1.89E-12	21995.00	30
	BBO	8.25E-13	1.79E-12	1090.00	30
f_8	EGSA	1.15E-05	1.14E-05	36455.00	30
	GSA	1.16E-05	1.17E-05	49801.67	30
	FBGSA	1.20E-05	1.22E-05	47368.33	30
	BBO	4.00E-01	3.26E-01	83296.67	18
f_9	EGSA	2.70E-14	3.90E-14	127580.00	30
	GSA	6.47E-02	6.67E-02	160010.00	28
	FBGSA	2.70E-14	5.34E-14	155450.00	30
	BBO	2.58E-14	3.10E-14	18711.67	30
f_{10}	EGSA	6.74E-06	6.33E-06	29600.00	30
	GSA	5.95E-06	5.50E-06	39686.67	30
	FBGSA	6.24E-06	6.29E-06	36780.00	30
	BBO	3.34E-16	1.22E+00	200000.00	0
f_{11}	EGSA	7.48E-06	8.88E-05	30623.33	30
	GSA	6.00E-06	8.96E-05	43526.67	30
	FBGSA	5.66E-06	8.89E-05	41330.00	30
	BBO	6.55E-06	8.96E-05	2600.00	30
f_{12}	EGSA	4.13E-14	9.53E-13	179236.67	30
	GSA	1.34E-12	1.34E-11	200000.00	0
	FBGSA	9.24E-14	8.73E-13	192680.00	30
	BBO	2.02E-03	5.94E-03	200000.00	0

Table 3. Comparison based on Mann-Whitney U rank test at significance level $\alpha = 0.05$ and mean function evaluations

Test problems	EGSA Vs GSA	EGSA Vs FBGSA	EGSA Vs BBO
f_1	+	+	+
f_2	+	+	+
f_3	+	+	+
f_4	+	+	-
f_5	+	+	+
f_6	+	+	+
f_7	+	+	-
f_8	+	+	+
f_9	+	+	-
f_{10}	+	+	+
f_{11}	+	+	-
f_{12}	+	+	+
Total number of + sign	12	12	08

Moreover, for comparison of examined algorithms, in form of consolidated achievement boxplots [2] study of AFE is carried out. Boxplot study efficiently describe the empirical circulation of data graphically. The boxplots for EGSA, GSA, FBGSA and BBO are depicted in Fig. 2. The results clearly show that interquartile range and medians of EGSA are relatively low.

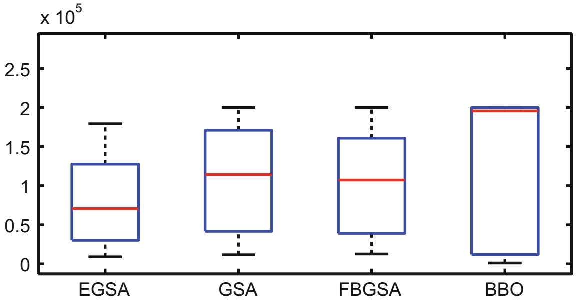


Fig. 2. Boxplots graphs (Average number of function evaluation)

To calculate the convergence speed of modified algorithm we use Acceleration Rate (AR) [9] which is represent as shown below:

$$AR = \frac{AFE_{compareAlgo}}{AFE_{EGSA}} \tag{19}$$

Table 4. Test Problems: TP, Acceleration Rate (AR) of *EGSA* compare to the Standard *GSA*, *FBGSA* and *BBO*

TP	GSA	FBGSA	BBO
f_1	1.212286342	1.105901417	1.568135487
f_2	2.378828427	2.378828427	2.378828427
f_3	1.199852621	1.096082356	1.319812587
f_4	1.287513894	1.40440904	0.594849944
f_5	1.284984628	1.21181692	2.975304969
f_6	1.920334864	2.700513097	2.581825547
f_7	1.470981115	1.519691386	0.075310917
f_8	1.366113473	1.299364513	2.284917478
f_9	1.254193447	1.218451168	0.146666144
f_{10}	1.340765766	1.242567568	6.756756757
f_{11}	1.421356264	1.349624469	0.08490258
f_{12}	1.115843112	1.075003255	1.115843112

here, $compareAlgo \in$ (GSA, FBGSA and BBO) and $AR > 1$ means EGSA is faster than compared algorithms. For investigate the AR of modified algorithm it compared with standard GSA, FBGSA and BBO, results of Table 2 are analyzed and the value of AR is calculated using Eq. 19. It is cleared from the Table 4 that convergence speed of EGSA is faster than other examined algorithms.

5 Conclusion

This paper presents a variant of GSA algorithm, known as Exploitative Gravitational Search Algorithm (EGSA). In the modified version, two control parameters gravitational constant G and $Kbest$ are modified. $Kbest$ is number of best individuals that exponentially decreases with the number of iterations increases and it increase the searching speed of algorithm. Gravitational constant G is also deceased with number of iterations increased and it enhance the convergence speed. This methodology is reliable and efficient and maintain the proper balance between the exploitation and exploration proficiency of the algorithm. The proposed algorithm is compared with GSA, FBGSA and BBO over different benchmark functions. The obtained results state that EGSA is a competitive variant of GSA and also a good choice for solving the continuous optimization problems. In future, the newly developed algorithm may be used to solve various real world optimization problems of continuous nature.

References

1. Bansal, J.C., Sharma, H., Arya, K.V., Deep, K., Pant, M.: Self-adaptive artificial bee colony. Optimization **63**(10), 1513–1532 (2014)
2. Bansal, J.C., Sharma, H., Arya, K.V., Nagar, A.: Memetic search in artificial bee colony algorithm. Soft Comput. **17**(10), 1911–1928 (2013)
3. Guo, Z.: A hybrid optimization algorithm based on artificial bee colony and gravitational search algorithm. Int. J. Digital Content Technol. Appl. **6**(17) (2012)
4. Gupta, A., Sharma, N., Sharma, H.: Fitness based gravitational search algorithm. In: Proceedings of IEEE International Conference on Computing Communication and Automation. IEEE (Accepted 2016)
5. Holliday, D., Resnick, R., Walker, J.: Fundamentals of physics (1993)
6. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)
7. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Bgsa: binary gravitational search algorithm. Nat. Comput. **9**(3), 727–745 (2010)
8. Sarafrazi, S., Nezamabadi-Pour, H., Saryazdi, S.: Disruption: a new operator in gravitational search algorithm. Scientia Iranica **18**(3), 539–548 (2011)
9. Sharma, K., Gupta, P.C., Sharma, H.: Fully informed artificial bee colony algorithm. J. Exp. Theor. Artif. Intell. 1–14 (2015)
10. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008)
11. Sudin, S., Nawawi, S.W., Faiz, A., Abidin, Z., Rahim, M.A.A., Khalil, K., Ibrahim, Z., Md Yusof, Z.: A modified gravitational search algorithm for discrete optimization problem