Fitness Varying Gravitational Constant in GSA

Jagdish Chand Bansal*, Susheel Kumar Joshi †
South Asian University, New Delhi, India

Atulya K. Nagar‡
Liverpool Hope University, UK

January 4, 2018

Received: date / Accepted: date

Abstract

Gravitational Search Algorithm (GSA) is a recent metaheuristic algorithm inspired by Newton's law of gravity and law of motion. In this search process, position change is based on the calculation of step size which depends upon a constant namely, Gravitational Constant (G). G is an exponentially decreasing function throughout the search process. Further, inspite of having different masses, the value of G remains same for each agent, which may cause inappropriate step size of agents for the next move, and thus leads the swarm towards stagnation or sometimes skipping the true optima.

To overcome stagnation, we first propose a gravitational constant having different scaling characteristics for different phase of the search process. Secondly, a dynamic behavior is introduced in this proposed gravitational constant which varies according to the fitness of the agents. Due to this behavior, the gravitational constant will be different for every agent based on its fitness and thus will help in controlling the acceleration and step sizes of the agents which further improve exploration and exploitation of the solution search space.

The proposed strategy is tested over 23 well-known classical benchmark functions and 11 shifted and biased benchmark functions. Various statistical analyses and a comparative study with original GSA, Chaos-based GSA (CGSA), Bio-geography Based Optimization (BBO) and DBBO has been carried out.

Keywords

Gravitational Search Algorithm (GSA), Swarm Intelligence, Gravitational Constant, Exploration, Exploitation.

1 Introduction

Gravitational search algorithm [13] is relatively new and very efficient optimization method belongs to the family of nature-inspired optimization algorithms. GSA is inspired by Newton's law of gravity and law of motion. The movement of agents (individuals) occurs under the influence of gravity forces [15]. Due to the gravity forces, a global movement generates which drives all agents towards the agents having heavier masses [6]. The details of the working of GSA are given in Section 2.

GSA has been modified in several ways to improve its performance. Inspired by Particle swarm optimization, Seyedali Mirjalili et al. [11] proposed a variant of GSA, namely PSOGSA in which each agent memorizes its previous best position. To improve the exploration and exploitation ability of GSA, Sarafrazi et al. [15] proposed a disruption operator. Doraghinejad et al. [4] improved the convergence characteristic of GSA, by introducing a new operator based on black hole phenomena. Seyedali Mirjalili et al. [12] improved the exploitation ability of GSA by incorporating the Gbest solution (best solution obtained so far) in the search strategy of GSA. To improve the

*jcbansal@gmail.com
†sushil4843@gmail.com
‡nagara@hope.ac.uk

convergence speed of GSA, Shaw et al. [17] initialized the swarm using opposition-based learning. Chen et al [3] introduced a hybrid GSA, in which a multi-type local improvement scheme is used as a local search operator. To improve the exploitation ability of GSA, Susheel et al. [7] introduced the encircle behavior of grey wolf in GSA.

In GSA, the concept of the dynamic (adaptive) parameter is proposed by Seyedali Mirjalili et al. [10]. In the proposed variant, the gravitational constant (G) adapts the chaotic behaviour using 10 different chaotic maps. For a fix chaotic map, G follows a fix chaotic nature throughout the search process. In [14], G is controlled by the fuzzy logic controller to improve the efficiency of GSA. In [1], design of experiment (DOE) method is used to tune the GSA parameters.

A proper balance between exploration and exploitation is required for an efficient nature-inspired algorithm. According to [19], good exploration ensures a thorough search in the search space while exploitation concentrates in the neighborhood of the best solution to ensure optimality. At the initial phase of search process the solutions may be far from the optimum solution, hence a large step size is required at the beginning (exploration) and when the solutions are converged towards an optimum solution, the small step size is needed for better exploitation in the neighborhood of the solution [8].

To improve the exploitation and exploration properties of the original gravitational search algorithm, a modified version of GSA called Fitness Varying Gravitational Constant in GSA (FVGGSA) is introduced in this paper. First, a gravitational constant having different scaling characteristics for the different phase of the search process is employed to avoid the possibility of stagnation in intermediate phases of search process. Further, each agent is incorporated with an ability to accelerate itself due to its individual gravitational constant which depends upon the fitness probability.

Therefore, both modifications have complementary advantages which provide a novel approach of self adapting step size for the next move towards the optimum, resulting in a balanced trade-off between exploration and exploitation properties of the algorithm.

To the best of the authors knowledge these kind of settings for gravitational constant which incorporate both fitness of the agent and different scaling parameters for different phases of the search process have not been proposed and implemented earlier in the literature. These two modifications make the proposed variant more efficient than the other previous variants of GSA in terms of dynamic parameters and novelty, respectively.

The remaining paper is organized as follows: Section 2 provides an overview of basic GSA. Fitness varying Gravitational Constant in GSA is proposed in Section 3. Section 4 describes the experiment results and comparative study. Finally the paper is concluded in section 5.

2 STANDARD GSA

Gravitational Search Algorithm (GSA) is a new swarm intelligence technique for optimization developed by Rashedi et al [13]. This algorithm is inspired by the law of gravity and the law of motion.

The GSA algorithm can be described as follows:
Consider the swarm of N agents, in which each agent X_i in the search space $\mathbb{S}$ is defined as:

$$X_i = (x_i^1, ..., x_i^d, ..., x_i^n), \quad \forall\, i = 1, 2, ..., N \tag{1}$$

Here, X_i shows the position of i^{th} agent in n-dimensional search space $\mathbb{S}$. The mass of each agent depends upon its fitness value as follows:

$$q_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{2}$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{N} q_j(t)}, \quad \forall\, i = 1, 2, ..., N \tag{3}$$

Here,
$fit_i(t)$ is the fitness value of agent X_i at iteration t,
$M_i(t)$ is the mass of agent X_i at iteration t.
Worst(t) and best(t) are worst and best fitness of the current population, respectively.

The acceleration of i^{th} agent in d^{th} dimension is denoted by $a_i^d(t)$ and defined as:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{4}$$

Where $F_i^d(t)$ is the total force acting on the i^{th} agent by a set of Kbest heavier masses in d^{th} dimension at iteration t. $F_i^d(t)$ is calculated as:

$$F_i^d(t) = \sum_{j \in KBEST, j \neq i} rand_j F_{ij}^d(t) \tag{5}$$

Here, KBEST is the set of first K agents with the best fitness values and biggest masses and $rand_j$ is a uniform random number between 0 and 1. Kbest is a linearly decreasing function of time. The value of Kbest will reduce in each iteration and at the end only one agent will apply force to the other agents. At the t^{th} iteration, the force applied on agent i from agent j in the d^{th} dimension is defined:

$$F_{ij}^d(t) = G(t)\frac{M_i(t)M_j(t)}{R_{ij} + \epsilon}(x_i^d(t) - x_j^d(t)) \tag{6}$$

Here, $R_{ij}(t)$ is the Euclidean distance between two agents, i and j. ϵ is a small number.
Finally, the acceleration of an agent in d^{th} dimension is calculated as:

$$a_i^d(t) = \sum_{j \in KBEST, j \neq i} rand_j G(t)\frac{M_j(t)}{R_{ij} + \epsilon}(x_i^d(t) - x_j^d(t)), \tag{7}$$

$d = 1, 2, ..., n$ and $i = 1, 2, ..., N$.
$G(t)$ is called gravitational constant and is a decreasing function of time:

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \tag{8}$$

G_0 and α are constants and set to 100 and 20, respectively. T is the total number of iterations.
The velocity update equation of an agent X_i in d^{th} dimension is given below:

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

Based on the velocity calculated in equation (9), the position of an agent X_i in d^{th} dimension is updated using position update equation as follow:

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

where $v_i^d(t)$ and $x_i^d(t)$ present the velocity and position of agent X_i in d^{th} dimension, respectively. $rand_i$ is uniform random number in the interval $[0, 1]$.

3 Fitness Varying Gravitational Constant in GSA

The robustness and effectiveness of a swarm based meta-heuristic algorithms depend upon the balance between exploration and exploitation capabilities [5]. In the initial iterations of the solution search process, exploration of search space is preferred. This can be obtained by allowing to attain large step sizes by agents during early iterations. In the later iterations, exploitation of search space is required to avoid the situation of skipping the global optima [16]. Thus the candidate solutions should have small step sizes for exploitation in later iterations.

According to the velocity update equation of GSA (equation (9)), acceleration plays a crucial role in balancing the exploration and exploitation. It is clear from equation (7) that the acceleration is a function of gravitational constant $G(t)$, masses $M_i(t)$ and distances R_{ij}. It is directly proportional to gravitational constant $G(t)$. For the higher value of $G(t)$ the acceleration will be higher hence step size will be larger, which causes exploration. Whereas the small value of $G(t)$ generates low acceleration and thus small step size in subsequent iterations will provide exploitation of the search space.

Therefore, the performance of GSA depends upon the gravitational constant $G(t)$ due to its role as a controller of step size for agent's movement. Mathematically, $G(t)$ is an exponentially decreasing function with respect to iterations by keeping scaling constant G_0 same throughout the search process. Due to this same value of scaling constant G_0, throughout the search process,

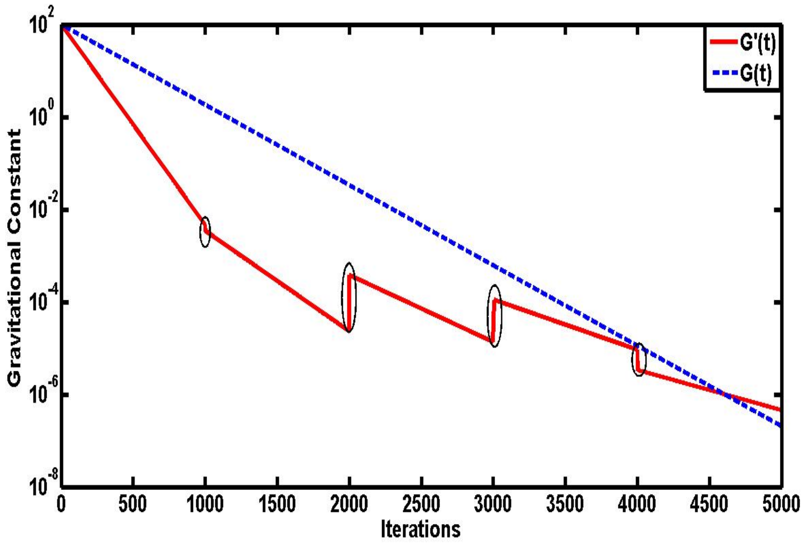


Figure 1: Original G(t) Vs proposed G'(t)

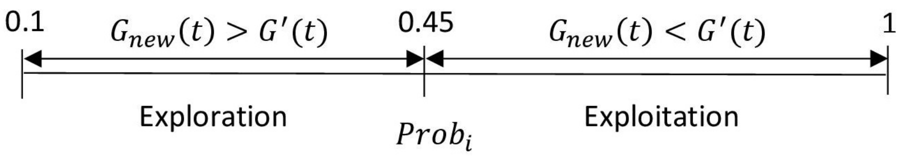


Figure 2: Relation between $G_{new}(t)$ and $G'(t)$

gravitational constant $G(t)$ does not significantly change over iterations. Therefore the step size of agent's movement also does not significantly change which further reduces the convergence speed of the algorithm.

To overcome this deficiency and make GSA faster, a new gravitational constant is introduced which have different scaling constants for different phase of the search process. A new gravitational constant is a concatenation of the different exponentially decreasing functions for different phases of the search process and defined as:

$$G'(t) = Ze^{-\alpha \frac{t}{\eta}} \tag{11}$$

where Z is scaling constant and is different for different phases of the search process. To apply the above defined gravitational constant, the entire search process is divided into phases of equal number of iterations, example 1000 (for this study) when the total number of iterations $T = 5000$. Based on numerical experiments on selected test problems the values of scaling constant for various phases are determined in Table 1. As expected, in initial phase, the value of Z is high, while in the last phase, it is minimum. In general, the value of Z for different phases of the search process can be obtained using function given below and is obtained by approximating the data of Table 1.

Table 1: Scaling constants for different phases of the search process

Z	Range (in iterations)
100	0-1000
0.5	1001-2000
0.3	2001-3000
0.2	3001-4000
0.01	4001-5000

$$Z(x) = 1408e^{-(0.00529)x} \tag{12}$$

Here x is the mid point of the considered range of the phase. For each Z, t is the current iteration and η is the maximum iteration of its corresponding range.
A comparison between original gravitational constant $G(t)$ and the proposed $G'(t)$ is shown in Fig. 1. In Fig. 1, circles represent the effect of different values of Z in $G'(t)$ with respect to different phases of the search process. At these points, clearly, the value of G' changes suddenly, which prohibits the search process for stagnation. Since the reducing constant α is responsible to navigate the search process from exploration to exploitation phase. This navigation provides the good convergence speed to GSA. Therefore, to make the faster GSA, α is set to 10 in equation (11).

Additionally, the requirement of exploration or exploitation can also be decided by the fitness of an agent. Since the low fitness implies that the agent is not near the optima, less fit agents can be recruited to explore the search space while high fit agents can be appointed to exploit their neighborhood. Therefore, a dynamic behavior of $G'(t)$ based on the fitness of agents is introduced. The proposed fitness varying gravitational constant is defined as:

$$G_{new}(t) = G'(t)(C - prob_i) \tag{13}$$

Here C is a constant and $prob_i$ is the probability related to i^{th} agent and calculated as below:

$$prob_i = \frac{0.9 \times fit(i)}{maxfit} + 0.1 \tag{14}$$

In this equation $fit(i)$ is the fitness value of i^{th} agent and maxfit is the maximum fitness of any agent in the current population. The fitness of an agent is calculated using the objective value as follow:

$$fit(i) = \begin{cases} 1 + abs(f_i), & \text{if } (f_i < 0) \\ \frac{1}{1+f_i}, & \text{if } (f_i \geq 0) \end{cases} \tag{15}$$

It is clear from equation (14) that the probability $prob_i$ is proportional to $fit(i)$. The GSA with proposed fitness varying gravitational constant is named as Fitness Varying Gravitational Constant GSA (FVGGSA).

From equation (7), $a(t) \propto G(t)$ which implies that $a(t) \propto G'(t)$, i.e. as $G'(t)$ increases, $a(t)$ and thus exploration capability of GSA increases. Thus in order to have a better exploration capability newly defined gravitational constant $G_{new}(t)$ should be larger than $G'(t)$. That is

$$G_{new}(t) > G'(t) \text{ or } G'(t) \times (C - prob_i) > G'(t)$$

$$\Rightarrow \quad (C - prob_i) > 1, \text{ (Since } G'(t) > 0)$$

$$\Rightarrow \quad C > 1 + prob_i$$

From equation (14), $0.1 \leq prob_i \leq 1$ and the average value of $prob_i$ is 0.45. Thus we set the value of constant C to be $1 + 0.45 = 1.45$.
Now if $C = 1.45$, then the proposed gravitational constant becomes:

$$G_{new}(t) = G'(t) \times (1.45 - prob_i) \tag{16}$$

It is clear from equation (16) that when $0.1 \leq prob_i < 0.45$, or when fitness is relatively worse, then $G_{new}(t) > G'(t)$, i.e. FVGGSA better explores when fitness has not reached at matured level. On the other hand, when $0.45 < prob_i \leq 1$ or when fitness is relatively better then $G_{new}(t) < G'(t)$, i.e. FVGGSA better exploits (Fig. 2). In case of $prob_i = 0.45$, which is very rare, $G_{new}(t) = G'(t)$. Finally, due to fitness dependent $G_{new}(t)$, the search process becomes explorative in early iterations while exploitative in later iterations. Fig. 3 illustrates the comparative behavior of $G_{new}(t)$ and $G(t)$ of an agent for benchmark functions f_3, f_{17}, f_{18}, f_{21}, f_{22} and f_{23} (refer section 4.1). It is clear that for most of the problems, $G_{new}(t) \geq G(t)$ in early iterations and $G_{new}(t) \leq G(t)$ for later iterations. The flow chart of so proposed FVGGSA is shown in Fig. 4.

4 Results and Discussion

4.1 Test bed under consideration

In this section, the proposed FVGGSA is tested over 23 test functions (test bed 1) and 11 shifted and biased test functions (test bed 2) [10]. The benchmark functions of test bed 1 and 2 are listed in Tables 2 and 3, respectively. In these Tables, *Search Range* denotes the domain of the function's search space, n indicates the dimension of function, C symbolizes the characteristics of benchmark functions and AE is the acceptable error.

The characteristics (C) of benchmark functions are classified into different categories like unimodal (U), multimodal (M), separable (S) and non-separable (N). Test bed 2 contains the benchmark functions having higher complexities due to their shift and bias nature.

4.2 Experimental setting

In order to validate the effectiveness and robustness of proposed algorithm, FVGGSA is compared with a recent GSA variant, namely Chaotic GSA (CGSA) [10]. In CGSA, there are 10 different variants (CGSA1 to CGSA10) based on 10 different chaotic maps. As per the original paper, CGSA8 and CGSA9 are the best two variants than others. Therefore, FVGGSA is compared with two best CGSA variants (CGSA8 and CGSA9) along with basic GSA, biogeography-based optimization (BBO) [18] and Disruption in biogeography-based optimization (DBBO) [2]. This comparison has been done over the test bed 1 with the popular experimental setting given in section 4.2.1.

In order to check the robustness of the proposed FVGGSA, it is further tested over more complex shifted and biased problems of test bed 2. To perform a fair comparison between FVGGSA and CGSA, the parameter setting for this test bed has been adopted from [10] as it is. The detailed description about the choice of the parameter settings can be found in [10]. The parameter setting for test bed 2 is given in section 4.2.2.

4.2.1 Parameter setting for test bed 1

- The number of simulations/run =30,
- Swarm size=50,
- The stopping criteria is either acceptable error (refer Table 2) has been achieved or maximum number of function evaluations (which is set to be 200000) is reached,
- Parameters for the algorithms GSA [13], BBO [18], DBBO [2], CGSA8 [10] and CGSA9 [10] are considered from the corresponding resources.

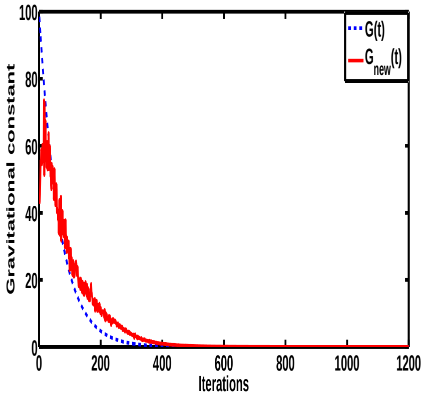
4.2.2 Parameter setting for test bed 2

- The number of simulations/run =20,
- Swarm size=30,
- The stopping criteria is the maximum number of function evaluations (which is set to be 20500) is reached,
- Parameters for all the variants of chaotic GSA [10] and GSA are adopted from their original papers.

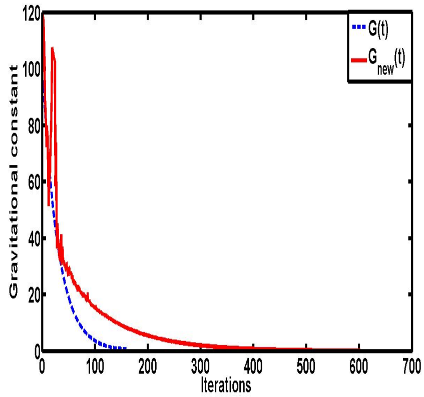
4.3 Result and statistical analysis of experiments

4.3.1 Test bed 1

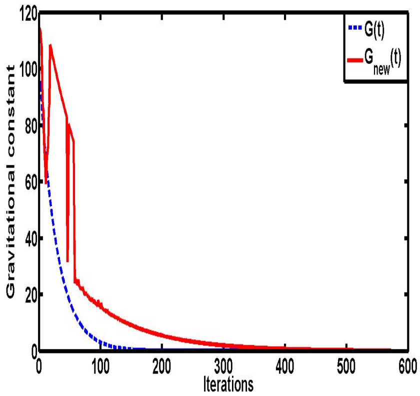
The results of the considered algorithms over the benchmark functions of test bed 1 are listed in Table 4. In this table, the criteria of comparison are standard deviation (*SD*), mean error (*ME*), average number of function evaluations (*AFEs*) along with the success rate (*SR*). *AFEs*, *SR* and *ME* present the efficiency, reliability and accuracy of an algorithm, respectively. The bold entries present the supremacy of an algorithm over others. Table 4 shows that most of the time FVGGSA



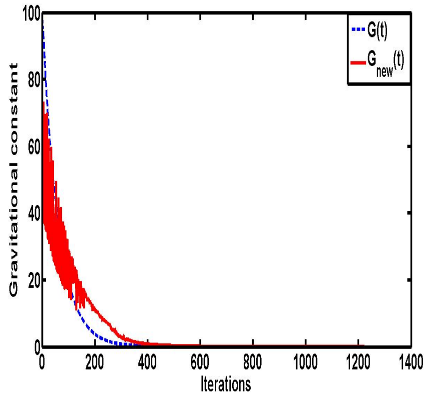
(a) Comparative behaviour for f_3



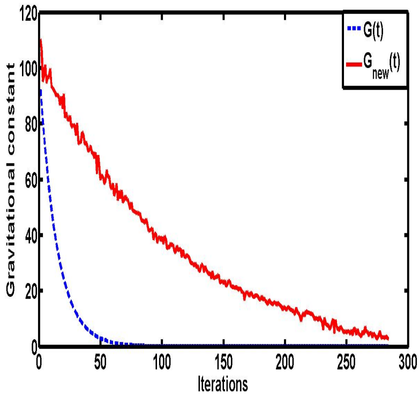
(b) Comparative behaviour for f_{17}



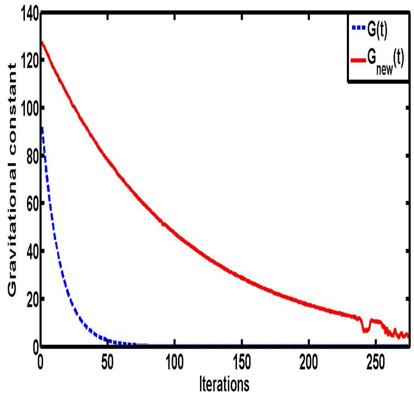
(c) Comparative behaviour for f_{18}



(d) Comparative behaviour for f_{21}



(e) Comparative behaviour for f_{22}



(f) Comparative behaviour for f_{23}

Figure 3: Comparative behavior of $G_{new}(t)$ and $G(t)$ of an agent for benchmark functions (mentioned in Table 2) f_3, f_{17}, f_{18}, f_{21}, f_{22} and f_{23}

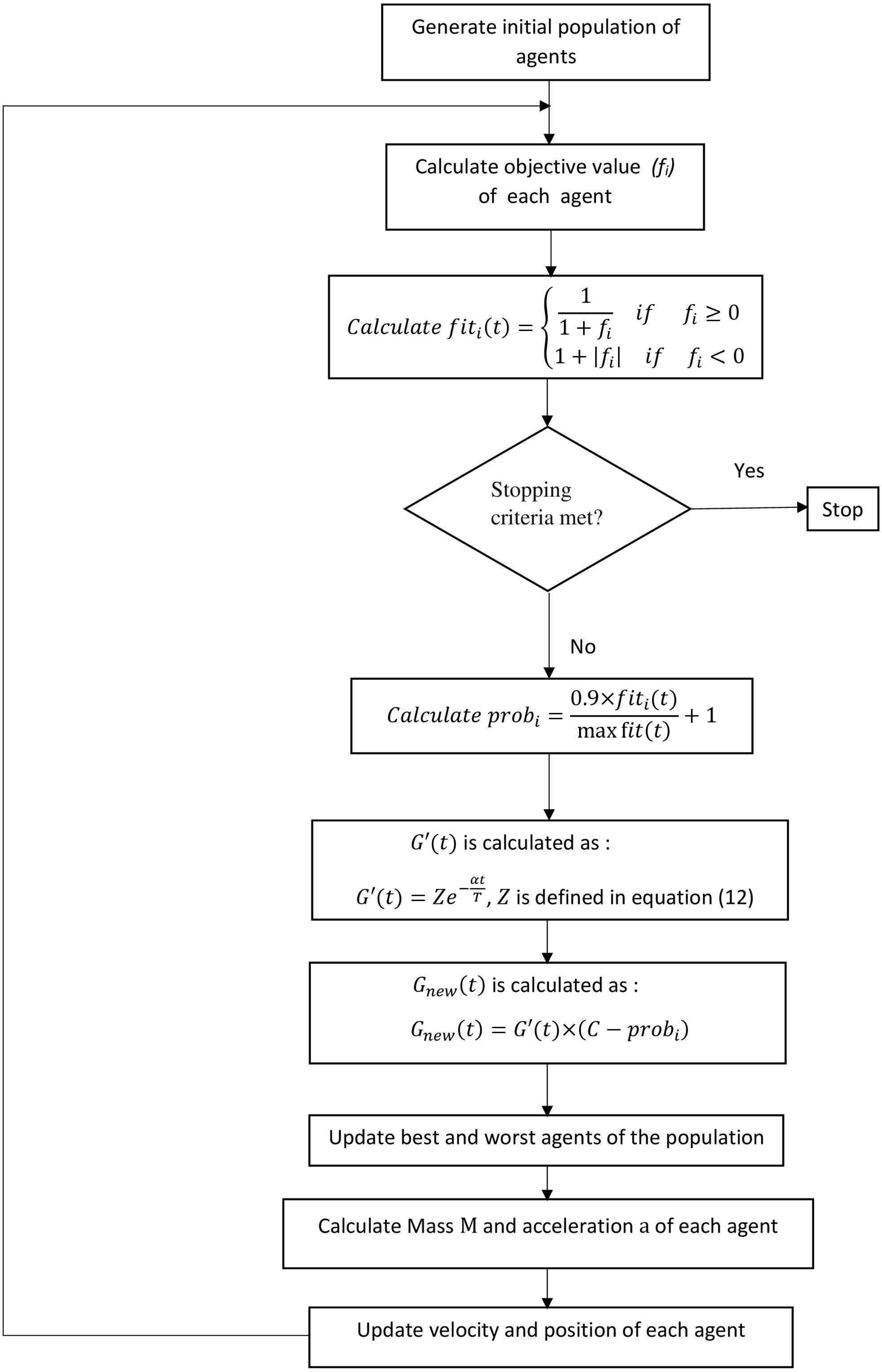


Figure 4: Flowchart of Fitness Varying Gravitational Constant in GSA

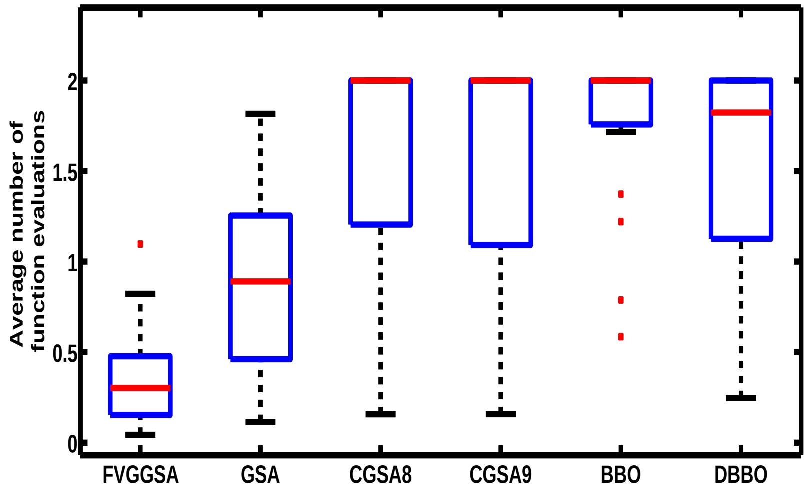


Figure 5: Boxplots (Analysis of average number of function evaluations for test bed 1)

dominates other algorithms with respect to efficiency, reliability and accuracy.

Further, to compare the algorithms on the basis of *AFEs*, boxplots analyses have been carried out. From Fig. 5, it is clear that the boxplot of FVGGSA have less interquartile range and medians as compared to GSA, CGSA8, CGSA9, BBO and DBBO which implies that FVGGSA is more efficient over other considered algorithms. This difference may occur due to chance and therefore data comparison test is required. It is clear from the Fig. 5 that the data used in the boxplot analysis are not normally distributed. Therefore a non-parametric statistical test, the Mann Whitney U rank sum test is applied.

Table 2: Benchmark functions (test bed 1)

Test problem	**Objective function**	**Search Range**	**Optimum Value**	**n**	**C**	**AE**
Sphere	$f_1(x)=\sum_{i=1}^{n} x_i^2$	[-5.12, 5.12]	$f(0)=0$	30	U,S	$1.00E-05$
De Jong f4	$f_2(x)=\sum_{i=1}^{n} i.(x_i)^4$	[-5.12 5.12]	$f(\vec{0})=0$	30	U,S	$1.00E-05$
Ackley	$f_3(x)=-20+e+exp(-\frac{0.2}{n}\sqrt{\sum_{i=1}^{n} {x_i}^3})-exp(\frac{1}{n}\sum_{i=1}^{n}\cos\,(2\pi x_i)x_i)$	[-30, 30]	$f(0)=0$	30	M,N	$1.0E-05$
Alpine	$f_4(x)=\sum_{i=1}^{n} \lvert x_i \sin\, x_i+0.1x_i\rvert$	[-10, 10]	$f(0)=0$	30	M,S	$1.0E-05$
Exponential	$f_5(x)=-(exp(-0.5\sum_{i=1}^{n} {x_i}^2))+1$	[-1, 1]	$f(0)=-1$	30	M,N	$1.0E-05$
brown3	$f_6(x)=\sum_{i=1}^{n-1}({x_i}^{2^{(x_{i+1})^2+1}}+{x_{i+1}}^{2^{{x_i}^2+1}})$	[-1, 4]	$f(0)=0$	30	U,N	$1.0E-05$
Schwefel 222	$f_7(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[-10, 10]	$f(0)=0$	30	U,N	$1.0E-05$
Axis parallel hyper-ellipsoid	$f_8(x)=\sum_{i=1}^{n} i.x_i^2$	[-5.12, 5.12]	$f(0)=0$	30	U,S	$1.0E-05$
Sum of different powers	$f_9(x)=\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	$[-1,1]$	$f(\vec{0})=0$	30	U,S	$1.0E-05$
Step function	$f_{10}(x)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	[-100 100]	$f(-0.5 \leq x \leq 0.5)=0$	30	U,S	$1.0E-05$
Rotated hyper-ellipsoid	$f_{11}(x)=\sum_{i=1}^{n}\sum_{j=1}^{i} x_j^2$	[-65.536, 65.536]	$f(0)=0$	30	U,N	$1.0E-05$
Levy montalvo 2	$f_{12}(x)=0.1(\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2\times(1+\sin^2(3\pi x_{i+1}))+(x_n-1)^2(1+\sin^2(2\pi x_n))$	[-5, 5]	$f(1)=0$	30	M,N	$1.0E-05$
Beale	$f_{13}(x)=[1.5-x_1(1-x_2)]^2+[2.25-x_1(1-x_2^2)]^2+$ $[2.625-x_1(1-x_2^3)]^2$	$[-4.5,4.5]$	$f(3,0.5)=0$	2	U,N	$1.0E-05$
Colville	$f_{14}(x)=100[x_2-x_1^2]^2+(1-x_1)^2+90(x_4-x_3^2)^2+(1-x_3)^2+10.1[(x_2-1)^2+(x_4-1)^2]+19.8(x_2-1)(x_4-1)$	[-10,10]	$f(\vec{1})=0$	4	M,N	$1.0E-05$
Branins's	$f_{15}(x)=a(x_2-bx_1^2+cx_1-d)^2+e(1-f)\cos x_1+e$	$-5\leq x_1\leq 10$, $0\leq x_2\leq 15$	$f(-\pi,12.275)$ $=0.3979$	2	M,N	$1.0E-05$
2D Tripod	$f_{16}(x)=p(x_2)(1+p(x_1))+\lvert(x_1+50p(x_2)(1-2p(x_1)))\rvert+\lvert(x_2+50(1-2p(x_2)))\rvert$ where $p(x)=1$ for $x\geq 0$	[-100,100]	$f(0,-50)=0$	2	M,N	$1.0E-04$
Shifted-parabola	$f_{17}(x)=\sum_{i=1}^{n} z_i^2+f_{bias}, z=x-o, x=[x_1,x_2,...,x_n], o=[o_1,o_2...,o_n]$	[-100, 100]	$f(o)=f_{bias}$ $=-450$	10	U,S	$1.00E-05$
Shifted-Schwefel 1.2	$f_{18}(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i} z_j)^2+f_{bias}$ $z=x-o, x=[x_1,x_2,...,x_n], o=[o_1,o_2...,o_n]$	[-100, 100]	$f(o)=f_{bias},$ $=-450$	10	U,N	$1.00E-05$
Gear train	$f_{19}(\vec{x})=\left(\frac{1}{6.931}-\frac{x_1x_2}{x_3x_4}\right)^2$	$[12,60]$	$f(19,16,43,49)$ $=2.7\times 10^{-12}$	4	—	$1.0E-15$
Six-hump camel back	$f_{20}(x)=(4-2.1x_1^2+x_1^4/3)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$	[-5, 5]	$f(-0.0898,0.7126)$ $=-1.0316$	2	M,N	$1.0E-05$
Easom's function	$f_{21}(x)=-\cos x_1 \cos x_2 e^{((-(x_1-\pi)^2-(x_2-\pi)^2))}$	[-100, 100]	$f(\pi,\pi)=-1$	2	U,N	$1.0E-13$
Hosaki Problem	$f_{22}(x)=(1-8x_1+7x_1^2-7/3x_1^3+1/4x_1^4)x_2^2\exp(-x_2)$ subject to $0\leq x_1\leq 5, 0\leq x_2\leq 6$	[0, 5], [0, 6]	-2.3458	2	M,N	$1.0E-05$
McCormick	$f_{23}(x)=\sin(x_1+x_2)+(x_1-x_2)^2-\frac{3}{2}x_1+\frac{5}{2}x_2+1$	$-1.5\leq x_1\leq 4$, $-3\leq x_2\leq 3$	f(-0.547,-1.547) $=-1.9133$	30	M,N	$1.0E-04$

Table 3: Shifted and biased benchmark functions (test bed 2) [10]

Objective function	**Search Range**	**Optimum Value**	**n**	**C**
$F_1(x) = \sum_{i=1}^{n}(x_i + 40)^2 - 80$	[-100, 100]	$f_{min} = -80$	30	U
$F_2(x) = \sum_{i=1}^{n} \|x_i + 7\| + \prod_{i=1}^{n} \|x_i + 7\| - 80$	[-10, 10]	$f_{min} = -80$	30	U
$F_3(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i}(x_j + 60))^2 - 80$	[-100, 100]	$f_{min} = -80$	30	U
$F_4(x) = max\left\{\|x_i + 60\|, 1 \leq i \leq n\right\} - 80$	[-100, 100]	$f_{min} = -80$	30	U
$F_5(x) = \sum_{i=1}^{n}\left(\|\left(x_i + 60\right) + 0.5\|\right)^2 - 80$	[-100, 100]	$f_{min} = -80$	30	U
$F_6(x) = \sum_{i=1}^{n} - \left(x_i + 300\right) sin\left(\sqrt{\|\left(x_i + 300\right)\|}\right)$	[-500, 500]	$f_{min} = -418.9829 \times (32)$	30	M
$F_7(x) = \sum_{i=1}^{n}\left[\left(x_i + 2\right)^2 - 10cos\left(2\pi\left(x_i + 2\right)\right) + 10\right] - 80$	[-5.12, 5.12]	$f_{min} = -80$	30	M
$F_8(x) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i + 20)^2}) - exp(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi(x_i + 20)\right)) + 20 + e - 80$	[-32, 32]	$f_{min} = -80$	30	M
$F_9(x) = \frac{1}{4000}\sum_{i=1}^{n}\left(x_i + 400\right)^2 - \prod_{i=1}^{n} cos\left(\frac{(x_i + 400)}{\sqrt{i}}\right) + 1 - 80$	[-600, 600]	$f_{min} = -80$	30	M
$F_{10}(x) = \frac{\pi}{n}\left\{10sin\left(\pi y_1\right) + \sum_{i=1}^{n-1}\left(y_i - 1\right)^2\left[1 + 10sin^2\left(\pi y_{i+1}\right)\right] + \left(y_n - 1\right)^2\right\} + \sum_{i=1}^{n} u\left(\left(x_i + 30\right), 10, 100, 4\right) - 80$, where, $y_i = 1 + \frac{(x_i + 30) + 1}{4}, u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m x_i > a \\ 0 - a < x_i < a \\ k\left(-x_i - a\right)^m x_i < -a \end{cases}$	[-50, 50]	$f_{min} = -80$	30	M
$F_{11}(x) = 0.1\left\{sin^2\left(3\pi\left(x_i + 30\right)\right) + \sum_{i=1}^{n}\left(\left(x_i + 30\right) - 1\right)^2\left[1 + sin^2\left(3\pi\left(x_i + 30\right) + 1\right)\right] + \left(\left(x_n + 30\right) - 1\right)^2\left[1 + sin^2\left(2\pi\left(x_n + 30\right)\right)\right]\right\} + \sum_{i=1}^{n} u\left(\left(x_i + 30\right), 5, 100, 4\right) - 80$	[-50,50]	$f_{min} = -80$	30	M

Table 4: Minimization results of test bed 1

TP	Algorithm	SD	ME	AFE	SR
f_1	FVGGSA	1.00545E-06	**8.56969E-06**	**32708.33333**	30
	GSA	1.06073E-06	8.72981E-06	95813.33333	30
	CGSA8	0.000326014	0.003587331	200000	0
	CGSA9	0.000296758	0.003083037	200000	0
	BBO	0.000381026	0.000816048	200000	0
	DBBO	**7.32694E-07**	9.56895E-06	173261.6667	29
f_2	FVGGSA	1.95728E-06	**7.30023E-06**	**18651.66667**	30
	GSA	1.52829E-06	8.17682E-06	62826.66667	30
	CGSA8	**9.44466E-07**	8.63861E-06	195010	30
	CGSA9	1.79769E-06	7.62802E-06	192921.6667	30
	BBO	1.65974E-06	8.11314E-06	122035	30
	DBBO	1.55575E-06	8.60549E-06	115993.3333	30
f_3	FVGGSA	6.92406E-07	**9.39163E-06**	**65520**	30
	GSA	**4.44256E-07**	9.4113E-06	160883.3333	30
	CGSA8	0.003942515	0.053256868	200000	0
	CGSA9	0.00376299	0.050965244	200000	0
	BBO	0.042374896	0.174415942	200000	0
	DBBO	0.003651462	0.012267751	200000	0
f_4	FVGGSA	4.07696E-07	9.49149E-06	**59978.33333**	30
	GSA	**4.00534E-07**	**9.40617E-06**	154651.6667	30
	CGSA8	0.001549659	0.023570243	200000	0
	CGSA9	0.001621385	0.023814067	200000	0
	BBO	0.001613291	0.009808612	200000	0
	DBBO	0.000325028	0.00076459	200000	0
f_5	FVGGSA	**8.007E-07**	**8.83583E-06**	**30910**	30
	GSA	8.1726E-07	8.96516E-06	91156.66667	30
	CGSA8	0.000172906	0.001765027	200000	0
	CGSA9	0.000146393	0.001551888	200000	0
	BBO	8.30753E-06	1.70658E-05	198063.3333	6
	DBBO	1.33938E-05	3.11127E-05	200000	0
f_6	FVGGSA	**7.17376E-07**	9.06744E-06	**34360**	30
	GSA	9.60908E-07	**8.918E-06**	99436.66667	30
	CGSA8	0.000737031	0.006326654	200000	0
	CGSA9	0.000718023	0.005931577	200000	0
	BBO	0.000152259	0.000363848	200000	0
	DBBO	9.38179E-07	9.513E-06	183240	28
f_7	FVGGSA	4.54212E-07	**9.20562E-06**	**82226.66667**	30
	GSA	**4.19962E-07**	9.45941E-06	181631.6667	30
	CGSA8	0.022057949	0.273234128	200000	0
	CGSA9	0.018211343	0.268694784	200000	0
	BBO	0.031361694	0.213726686	200000	0
	DBBO	0.001492103	0.002438375	200000	0
f_8	FVGGSA	9.83386E-07	**8.82813E-06**	**38831.66667**	30
	GSA	**7.32591E-07**	9.0581E-06	109796.6667	30
	CGSA8	0.003463022	0.027433071	200000	0
	CGSA9	0.00249644	0.031423683	200000	0
	BBO	0.005047	0.013401429	200000	0
	DBBO	2.57678E-05	2.89363E-05	198366.6667	7

Table 4 Continued:

TP	Algorithm	SD	ME	AFE	SR
f_9	FVGGSA	2.41338E-06	**6.44091E-06**	**10725**	30
	GSA	**2.09E-06**	7.39476E-06	46590	30
	CGSA8	2.59442E-06	7.07505E-06	127248.3333	30
	CGSA9	2.2068E-06	6.71658E-06	113008.3333	30
	BBO	0.000503803	0.00069678	200000	0
	DBBO	5.28231E-05	4.70202E-05	185148.3333	5
f_{10}	FVGGSA	0	0	**4316.666667**	30
	GSA	0	0	11375	30
	CGSA8	0	0	15656.66667	30
	CGSA9	0	0	15686.66667	30
	BBO	0	0	58551.66667	30
	DBBO	0	0	24601.66667	30
f_{11}	FVGGSA	1.01012E-06	**8.87152E-06**	**32681.66667**	30
	GSA	**6.45608E-07**	9.2917E-06	95475	30
	CGSA8	0.000309836	0.003219302	200000	0
	CGSA9	0.000251015	0.003258639	200000	0
	BBO	0.053371002	0.147479446	200000	0
	DBBO	0.00072175	0.00172012	200000	0
f_{12}	FVGGSA	**1.0255E-06**	**8.59766E-06**	**32926.66667**	30
	GSA	1.08817E-06	8.8444E-06	95508.33333	30
	CGSA8	0.000429894	0.003092997	200000	0
	CGSA9	0.000319062	0.003153544	200000	0
	BBO	0.000375113	0.00088644	200000	0
	DBBO	1.04111E-06	9.31676E-06	181406.6667	29
f_{13}	FVGGSA	2.74244E-06	4.93008E-06	**21693.33333**	30
	GSA	2.69049E-06	5.33222E-06	71498.33333	30
	CGSA8	3.01076E-06	4.80294E-06	166498.3333	30
	CGSA9	**2.67967E-06**	**4.001E-06**	156311.6667	30
	BBO	0.148906748	0.038065174	197476.6667	1
	DBBO	0.023836315	0.008202032	195848.3333	1
f_{14}	FVGGSA	**0.019985001**	**0.006282989**	**109688.3333**	**28**
	GSA	0.103028613	0.038774868	141136.6667	26
	CGSA8	0.036267406	0.052121955	200000	0
	CGSA9	0.00644022	0.030390784	199965	1
	BBO	6.446884307	5.502358837	200000	0
	DBBO	0.365423535	0.470397766	200000	0
f_{15}	FVGGSA	**2.80355E-05**	5.20081E-05	**14236.66667**	30
	GSA	2.87226E-05	5.04129E-05	39410	30
	CGSA8	3.14604E-05	5.38765E-05	80011.66667	30
	CGSA9	3.15725E-05	**4.38475E-05**	81398.33333	30
	BBO	0.001719436	0.001264256	174325	7
	DBBO	0.000508038	0.000459227	174518.3333	8
f_{16}	FVGGSA	**1.84578E-07**	6.92086E-07	**56578.33333**	30
	GSA	2.3E-07	**6.42095E-07**	146150	30
	CGSA8	0.000466421	0.009345424	200000	0
	CGSA9	0.00029439	0.010368093	200000	0
	BBO	0.065272948	0.085459647	200000	0
	DBBO	0.034748177	0.040469247	200000	0

Table 4 Continued:

TP	Algorithm	SD	ME	AFE	SR
f_{17}	FVGGSA	**1.42435E-06**	7.82029E-06	**29450**	30
	GSA	1.45544E-06	**7.75472E-06**	86926.66667	30
	CGSA8	0.000101188	0.000791677	200000	0
	CGSA9	9.99422E-05	0.001020917	200000	0
	BBO	0.044385826	0.071491278	200000	0
	DBBO	2.83246E-05	1.36422E-05	83506.66667	27
f_{18}	FVGGSA	**1.34021E-06**	**7.89446E-06**	**29495**	30
	GSA	1.64223E-06	8.35732E-06	86808.33333	30
	CGSA8	0.000103541	0.000735191	200000	0
	CGSA9	7.16396E-05	0.000918579	200000	0
	BBO	0.038002814	0.070375662	200000	0
	DBBO	2.58167E-05	1.80428E-05	109188.3333	23
f_{19}	FVGGSA	8.54501E-13	1.78991E-12	**6838.333333**	30
	GSA	7.49569E-13	**1.60954E-12**	21175	30
	CGSA8	8.18347E-13	1.79002E-12	30401.66667	30
	CGSA9	**7.16974E-13**	1.71E-12	29901.66667	30
	BBO	3.69823E-10	2.50834E-10	188806.6667	3
	DBBO	1.13766E-11	5.91029E-12	90226.66667	22
f_{20}	FVGGSA	**9.6028E-06**	**9.18315E-06**	**18121.66667**	30
	GSA	1.07195E-05	1.23835E-05	48965	30
	CGSA8	1.06159E-05	1.3273E-05	116328.3333	30
	CGSA9	9.49062E-06	8.80391E-06	114850	30
	BBO	0.000283252	0.000154304	171593.3333	10
	DBBO	0.000147182	8.99139E-05	135356.6667	16
f_{21}	FVGGSA	**2.92902E-14**	**4.35207E-14**	**66075**	**30**
	GSA	0.017531045	0.033333333	161190	29
	CGSA8	0.071910972	0.100106195	200000	0
	CGSA9	0.247396831	0.133426362	200000	0
	BBO	0.454861839	0.305210901	200000	0
	DBBO	0.297851511	0.108000333	200000	0
f_{22}	FVGGSA	**5.39755E-06**	4.82709E-06	**14253.33333**	30
	GSA	5.55481E-06	**4.60393E-06**	42300	30
	CGSA8	5.97998E-06	5.29309E-06	104328.3333	30
	CGSA9	6.32497E-06	5.75214E-06	84918.33333	30
	BBO	1.43222E-05	8.88844E-06	78816.66667	26
	DBBO	0.632842387	0.127965301	78606.66667	29
f_{23}	FVGGSA	7.05128E-06	8.71941E-05	**14365**	30
	GSA	6.77996E-06	**8.57392E-05**	44783.33333	30
	CGSA8	6.58545E-06	9.09028E-05	114931.6667	30
	CGSA9	**6.43314E-06**	8.92693E-05	103860	30
	BBO	2.61814E-05	0.000101775	137310	21
	DBBO	0.0176558	0.003367038	97730	26

The Mann-Whitney U rank sum test [9] is a non-parametric test for comparison among the data which are not normally distributed. In this study, this test is performed at 5% level of significance ($\alpha = 0.05$) with null hypothesis, 'There is no significant difference in the data', between FVGGSA-GSA, FVGGSA-CGSA8, FVGGSA-CGSA9, FVGGSA-BBO and FVGGSA-DBBO. If the significant difference between two data sets does not occur, it implies that the null hypothesis is accepted, therefore sign '=' appears. On the contrary, when the null hypothesis is rejected, '-' or '+' signs appears.

In this paper, the data sets are the *AFEs* of a particular algorithm. '-' or '+' sign shows that a particular algorithm has more or less number of function evaluations as compared to other. Table 5 presents the results of Mann-Whitney U rank sum test for *AFEs* of 30 runs. In Table 5, 114 '+' signs out of 115 comparisons assure that FVGGSA requires less number of function evaluations as compared to the other considered algorithms.

Table 5: Comparison based on the *AFEs* of 30 runs for test bed 1 using Mann Whitney U rank sum test at $\alpha = 0.05$ significance level

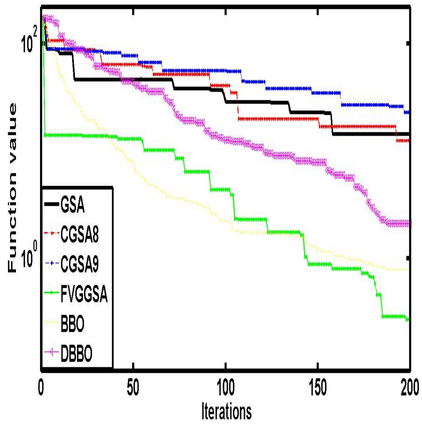
Test Problem	U rank sum test with FVGGSA				
	GSA	CGSA8	CGSA9	BBO	DBBO
f_1	+	+	+	+	+
f_2	+	+	+	+	+
f_3	+	+	+	+	+
f_4	+	+	+	+	+
f_5	+	+	+	+	+
f_6	+	+	+	+	+
f_7	+	+	+	+	+
f_8	+	+	+	+	+
f_9	+	+	+	+	+
f_{10}	+	+	+	+	+
f_{11}	+	+	+	+	+
f_{12}	+	+	+	+	+
f_{13}	+	+	+	+	+
f_{14}	=	+	+	+	+
f_{15}	+	+	+	+	+
f_{16}	+	+	+	+	+
f_{17}	+	+	+	+	+
f_{18}	+	+	+	+	+
f_{19}	+	+	+	+	+
f_{20}	+	+	+	+	+
f_{21}	+	+	+	+	+
f_{22}	+	+	+	+	+
f_{23}	+	+	+	+	+

To further verify the exploitation of FVGGSA, the convergence behavior of the considered algorithms over some unimodal and multimodal benchmark functions is illustrated in Fig. 6. It can be observed in Fig. 6, FVGGSA outperforms others in terms of exploitation ability due to its fastest convergence rate.

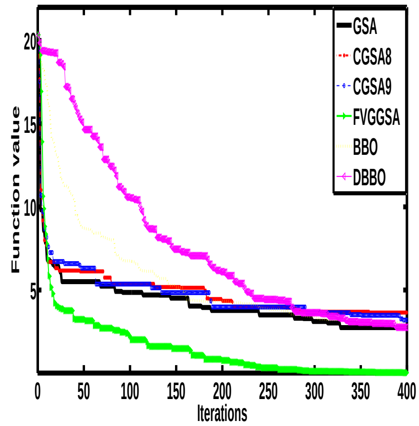
4.3.2 Test bed 2

To check the performance of the proposed algorithm over more complex problems, FVGGSA is re-evaluated over the shifted and biased benchmark problems of test bed 2. Table 6 and Table 7 present the experimental results which are obtained by the average of 20 independent runs. Except FVGGSA, other results are adopted from [10]. The criteria of comparison are mean and standard deviation (*SD*) of the objective function values. The bold entries indicate the best results. As per the results shown in Table 6 and Table 7, FVGGSA outperforms for 2 unimodal (F_1 and F_2) as well as 3 multimodal (F_6, F_7 and F_{10}) functions over other considered algorithms. For 3 functions (F_4, F_9 and F_{11}) FVGGSA is better than others except CGSA9. For F_5, FVGGSA is better than others except CGSA8 and CGSA9. While for F_3, FVGGSA is better than GSA only. Furthermore, to investigate the convergence speed of the proposed algorithm, FVGGSA is compared with GSA and the best CGSA variant under considered unimodal (F_1 and F_2) and multimodal (F_7 and F_{10}) benchmark functions. The convergence graphs are depicted in Fig. 7. It can be clearly observed that FVGGSA has the fastest convergence rate as compared to GSA and the best variant of CGSA.

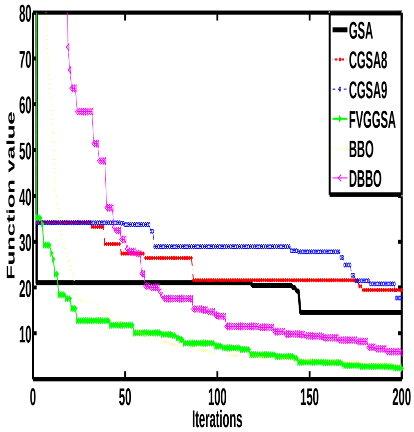
Based on the numerical results of FVGGSA on the problems of Test bed 1 and Test bed 2 it is suggested that FVGGSA can be applied to solve the problems in continuous domain which are non-separable and uni-modal or multi-modal.



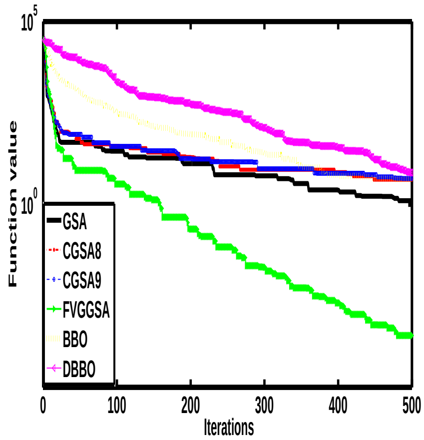
(a) Benchmark function f_1



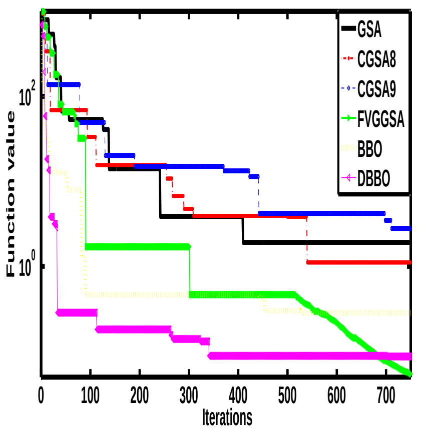
(b) Benchmark function f_3



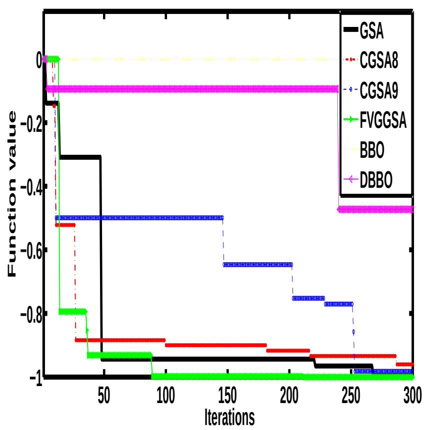
(c) Benchmark function f_7



(d) Benchmark function f_{11}



(e) Benchmark function f_{14}



(f) Benchmark function f_{21}

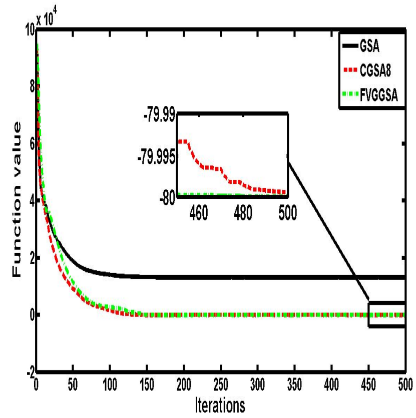
Figure 6: Convergence graph for benchmark functions f_1, f_3, f_7, f_{11}, f_{14} and f_{21}

Table 6: Minimization results of test bed 2

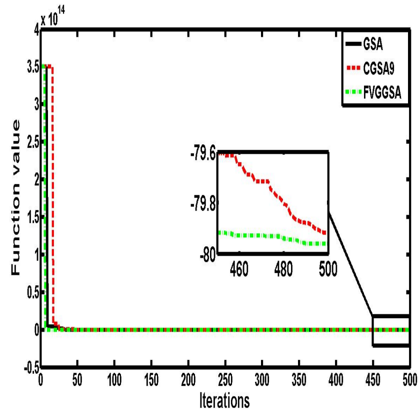
TP	Algorithm	Mean	SD	TP	Algorithm	Mean	SD
	GSA	9154.139	2816.071		GSA	-12.8803	21.443822
	CGSA1	-79.9991	0.00054		CGSA1	-79.8521	0.051772
	CGSA2	-79.999	0.000801		CGSA2	-79.8545	0.0445217
	CGSA3	2744.044	1557.386		CGSA3	-77.6142	6.736717
	CGSA4	-79.9987	0.000518		CGSA4	-79.8401	0.040522
F_1	CGSA5	-79.9981	0.001215	F_2	CGSA5	-79.7979	0.059315
	CGSA6	-79.999	0.00064		CGSA6	-79.875	0.026707
	CGSA7	-79.9985	0.001517		CGSA7	-79.8567	0.041229
	CGSA8	-79.9996	0.000112		CGSA8	-79.8856	0.036908
	CGSA9	-79.9995	0.000252		CGSA9	-79.8979	0.015186
	CGSA10	-79.9986	0.001537		CGSA10	-79.8319	0.025325
	FVGGSA	**-80**	**8.27402E-07**		FVGGSA	**-79.9909**	**0.000834806**
	GSA	8992532	1201459		GSA	-20.2138	3.804462
	CGSA1	116840.3	70535.26		CGSA1	-30.4488	2.30128
	CGSA2	198815.5	72140.38		CGSA2	-29.7255	2.330251
	CGSA3	3858229	1202078		CGSA3	-22.1425	3.010063
	CGSA4	162491.1	78431.77		CGSA4	-32.0524	2.912228
F_3	CGSA5	116130.9	63272.89	F_4	CGSA5	-29.1978	2.989176
	CGSA6	121970.3	57720.59		CGSA6	-29.8382	3.101845
	CGSA7	218939.4	115677.8		CGSA7	-29.7609	2.868508
	CGSA8	40212.85	23173.42		CGSA8	-29.3983	2.015278
	CGSA9	**17322.05**	**9866.881**		CGSA9	**-35.4132**	**2.487503**
	CGSA10	143840	99020.52		CGSA10	-29.9936	2.515029
	FVGGSA	6192650.299	1978710.505		FVGGSA	-32.34019838	3.314337773
	GSA	36385.55	5403.108		GSA	-5061.91	789.3759
	CGSA1	1417.568	717.114		CGSA1	-5543.91	821.204
	CGSA2	2942.762	1283.694		CGSA2	-5226.26	887.7445
	CGSA3	25172.97	3233.377		CGSA3	-5170.38	673.8648
	CGSA4	1996.549	1777.576		CGSA4	-5318.82	807.7437
F_5	CGSA5	1996.222	1330.493	F_6	CGSA5	-5206.9	755.3401
	CGSA6	1539.554	1224.341		CGSA6	-5213.43	848.2368
	CGSA7	2761.892	1495.06		CGSA7	-5375.46	685.5158
	CGSA8	158.2152	241.4738		CGSA8	-5724.74	888.7908
	CGSA9	**-79.9995**	**0.000347**		CGSA9	-6489.32	849.6746
	CGSA10	1478.906	789.3842		CGSA10	-5405.45	661.9886
	FVGGSA	1158.959481	1082.271164		FVGGSA	**-6899.186412**	**932.0734707**
	GSA	-19.2075	16.01267		GSA	-62.5807	1.69382
	CGSA1	-8.85697	19.27564		CGSA1	-76.4301	7.098432
	CGSA2	-2.92268	19.15199		CGSA2	-74.4996	8.191298
	CGSA3	-35.0574	14.54588		CGSA3	-64.6326	4.353233
	CGSA4	1.48412	19.66447		CGSA4	-73.0146	8.699273
F_7	CGSA5	-20.0007	8.708038	F_8	CGSA5	-73.4597	7.760572
	CGSA6	-5.91409	15.37048		CGSA6	-73.2779	7.991709
	CGSA7	-15.9009	17.35564		CGSA7	-78.3286	3.922819
	CGSA8	-3.01164	25.69204		CGSA8	-79.98	0.009413
	CGSA9	20.76884	37.64664		CGSA9	-76.2319	6.765571
	CGSA10	-2.05186	19.91167		CGSA10	-79.7545	0.694395
	FVGGSA	**-39.6039594**	**9.482934705**		FVGGSA	-60.88875666	0.427750195
	GSA	895.6395	115.0452		GSA	869659.2	691767.8
	CGSA1	809.924	69.23016		CGSA1	-50.3626	7.378042
	CGSA2	831.9158	134.318		CGSA2	-35.6267	10.94971
	CGSA3	910.7538	88.02444		CGSA3	475.8308	1401.28
	CGSA4	820.3869	68.68291		CGSA4	-47.1339	6.533372
F_9	CGSA5	828.7685	80.04076	F_{10}	CGSA5	-51.293	6.493393
	CGSA6	812.7824	80.71501		CGSA6	-41.9416	8.575967
	CGSA7	863.9838	68.1401		CGSA7	-48.5795	7.453836
	CGSA8	801.8709	104.4999		CGSA8	-58.2807	9.238928
	CGSA9	772.133	67.10897		CGSA9	-53.6841	4.819881
	CGSA10	854.8892	93.37514		CGSA10	-48.4999	8.102045

Table 7: Minimization results of test bed 2

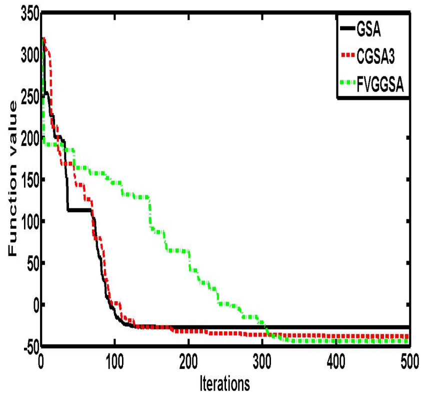
TP	Algorithm	Mean	SD
	GSA	14452359	17545762
	CGSA1	-78.4668	2.147558
	CGSA2	-78.2219	1.482622
	CGSA3	31138.24	43257.47
	CGSA4	-79.3375	0.808498
F_{11}	CGSA5	-79.2512	0.846976
	CGSA6	-79.4858	0.709402
	CGSA7	-74.0584	4.585205
	CGSA8	-79.8951	0.284321
	CGSA9	-79.9989	0.003469
	CGSA10	-79.5537	0.838046
	FVGGSA	-79.9078	0.29254664



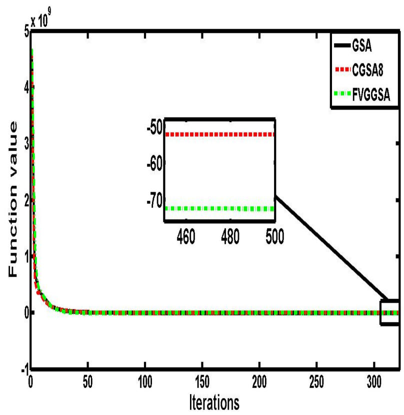
(a) Comparative behaviour for F_1



(b) Comparative behaviour for F_2



(c) Comparative behaviour for F_7



(d) Comparative behaviour for F_{10}

Figure 7: Convergence curves for shifted and biased benchmark functions

5 Conclusion

To avoid the possibility of stagnation in the search process, we first proposed a gravitational constant having different scaling characteristics for different phases of the search space. Next, a novel concept of fitness varying strategy is introduced in the above proposed gravitational constant. This behavior controls acceleration of the agents in such a way that the chance of skipping the global optima is reduced while maintaining the diversity. This self-accelerative behavior gives a special intelligence to each agent for choosing the appropriate step size for its next move. Through intensive experiments and analyses over 23 well-known benchmark functions and 11 shifted and biased benchmark functions, the proposed algorithm has proved its efficiency for unimodal as well as multimodal problems of continuous search space. Further, it is a good choice for non separable continuous problems also.

Acknowledgements: The second author acknowledges the funding from South Asian University New Delhi, India to carry out this research.

References

[1] M. Amoozegar and E. Rashedi. Parameter tuning of gsa using doe. In *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 431–436, Oct 2014.

[2] Jagdish Chand Bansal and Pushpa Farswan. A novel disruption in biogeography-based optimization with application to optimal power flow problem. *Applied Intelligence*, 46(3):590–615, 2017.

[3] Huiqin Chen, Sheng Li, and Zheng Tang. Hybrid gravitational search algorithm with random-key encoding scheme combined with simulated annealing. *IJCSNS*, 11(6):208, 2011.

[4] Mohammad Doraghinejad and Hossein Nezamabadi-pour. Black hole: A new operator for gravitational search algorithm. *International Journal of Computational Intelligence Systems*, 7(5):809–826, 2014.

[5] Agoston E Eiben and CA Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4):35–50, 1998.

[6] David Holliday, Robert Resnick, and Jearl Walker. Fundamentals of physics. 1993.

[7] Susheel Joshi and Jagdish Chand Bansal. Grey wolf gravitational search algorithm. In *Computational Intelligence (IWCI), International Workshop on*, pages 224–231. IEEE, 2016.

[8] Mohammad Khajehzadeh, Mohd Raihan Taha, Ahmed El-Shafie, and Mahdiyeh Eslami. A modified gravitational search algorithm for slope stability analysis. *Engineering Applications of Artificial Intelligence*, 25(8):1589–1597, 2012.

[9] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[10] Seyedali Mirjalili and Amir H Gandomi. Chaotic gravitational constants for the gravitational search algorithm. *Applied Soft Computing*, 2017.

[11] Seyedali Mirjalili and Siti Zaiton Mohd Hashim. A new hybrid psogsa algorithm for function optimization. In *Computer and information application (ICCIA), 2010 international conference on*, pages 374–377. IEEE, 2010.

[12] Seyedali Mirjalili and Andrew Lewis. Adaptive gbest-guided gravitational search algorithm. *Neural Computing and Applications*, 25(7-8):1569–1584, 2014.

[13] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

[14] F. s. Saeidi-Khabisi and E. Rashedi. Fuzzy gravitational search algorithm. In *2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)*, pages 156–160, Oct 2012.

[15] S Sarafrazi, H Nezamabadi-Pour, and S Saryazdi. Disruption: a new operator in gravitational search algorithm. *Scientia Iranica*, 18(3):539–548, 2011.

[16] Harish Sharma, Jagdish Chand Bansal, and K. V. Arya. Fitness based differential evolution. *Memetic Computing*, 4(4):303–316, 2012.

[17] Binod Shaw, V Mukherjee, and SP Ghoshal. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. *International Journal of Electrical Power & Energy Systems*, 35(1):21–33, 2012.

[18] Dan Simon. Biogeography-based optimization. *Evolutionary Computation, IEEE Transactions on*, 12(6):702–713, 2008.

[19] Xin-She Yang. *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.