Noname manuscript No.
(will be inserted by the editor)

A novel neighbourhood archives embedded gravitational constant in GSA

Susheel Kumar Joshi · Anshul Gopal · Shitu Singh · Atulya K. Nagar · Jagdish Chand Bansal

Received: date / Accepted: date

Abstract Due to its effective search mechanism, gravitational search algorithm (GSA) has became very popular and robust tool for the global optimization in a very short span of time. The search mechanism of GSA is based on its two features, namely K_{best} archive and gravitational constant G. The K_{best} archive stores the best agents (solutions) at any evolutionary state and hence helps GSA in search globally. Each agent interacts with exactly same agents of K_{best} archive without considering its current impact on the search process, results, a rapid loss of diversity, premature convergence and the high time complexity in GSA model. On the other hand, the exponentially decreasing behavior of G scales the step size of the agent. However, this scaling is same for all agents which may cause inappropriate step size for their next move, and thus leads the swarm towards stagnation or sometimes skipping the true optima. To address these problems, an improved version of GSA called 'A novel neighbourhood archives embedded gravitational constant in GSA (NAGGSA)' is proposed in this paper. In NAGGSA, we first propose two novel neighbourhood archives for each agent which helps in increased diversified search with less time complexity. Secondly, a novel gravitational constant is proposed for each agent according to the distance-fitness based scaling mechanism. The performance of the proposed variant is tested over different suites of well-known benchmark test functions. Experimental results and statistical analyses reveal that NAGGSA remarkably outperforms the compared algorithms.

Keywords Neighbourhood archive · Gravitational Search Algorithm (GSA) · Gravitational Constant · Meta-heuristics · Swarm Intelligence · Nature inspired optimization

Susheel Kumar Joshi
South Asian University, New Delhi, India
E-mail: sushil4843@gmail.com

Anshul Gopal
South Asian University, New Delhi, India
E-mail: anshulgopal@students.sau.ac.in

Shitu Singh
South Asian University, New Delhi, India
E-mail: singh.shitu93@gmail.com

Atulya K. Nagar
Liverpool Hope University, UK
E-mail: nagara@hope.ac.uk

Jagdish Chand Bansal
South Asian University, New Delhi, India
E-mail: jcbansal@sau.ac.in

1 Introduction

It is required to develop a mechanism that effectively makes a proper balance between exploration and exploitation in GSA. Exploration is the ability to produce a highly randomized behavior in the mechanism of an algorithm such that its candidate solutions (agents) explore the wide regions of the search space, whereas exploitation produces the neighbourhood search mechanism for the algorithm in which agents refine the promising regions of the search space. The performance metrics of an algorithm like efficiency, reliability, accuracy and the convergence speed greatly depend upon the trade-off between exploration and exploitation. For the best performance of these metrics, an algorithm should perform exploration in the early stages and execute refined exploitation in the latter stages of the search process. However, how to achieve a proper balance between these two remains an unsolved challenge [32].

The gravitational search algorithm (GSA) is a recent and very robust meta-heuristic algorithm inspired by the gravity rules [21]. In GSA, the social interaction among agents is guided by K_{best} archive which stores the superior agents of the current evolutionary state. Each agent interacts with the agents of the K_{best} archive to get the diverse knowledge of the different directions of the search space. Moreover, by lapse of time, the size of K_{best} archive linearly decreases from N (population size) to 1, results, a search mechanism that explores the search space in early stages and exploits in the later stages of the search process. Although the K_{best} archive in GSA provides a good trade-off between exploration and exploitation, it has the major problem due to the presence of global neighbourhood concept. Throughout the search process, agents learn from the same elites all the time. If the elites stagnate somewhere in the local optima, all agents may stagnate around this pseudo optimal region, resulting a premature convergence. Additionally, the large size of K_{best} in the early stages of the search process increases the time complexity of the GSA model, whereas the small size of K_{best} in the later stages provides a less knowledge about the search process, which inevitably causes quick loss of search diversity. Except K_{best} archive, Gravitational constant G is the second most important entity in GSA model which deals in the trade-off between exploration and exploitation by scaling the step size of the agents. Basically, G is the exponential decreasing function of time having two constant parameters G_0 and α. Through this deceasing behavior, exploration fades out and the exploitation turns to fade in. However, due to constants G_0 and α, G does not change significantly according to the search requirement, especially in the middle phase of the search process. In addition, in spite of having different masses, the value of G remains the 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 address the aforementioned drawbacks of GSA, many GSA variants have been developed by embedding new learning strategies into it. Mirjalili et al. [14] assigned a memory to each agent of GSA to improve the search ability. Sarafrazi et al. [23] improved the exploration and exploitation ability of GSA by using disruption operator. To overcome the premature convergence, Li et al. [10] hybridized differential evolution (DE) with GSA. To improve the exploitation, Chen et al [3] proposed a local search operator as a multi-type local improvement scheme. To improve the convergence speed of GSA, Shaw et al. [24] used the opposition-based learning. To prevent the premature convergence and improve the convergence characteristic of GSA, Doraghinejad et al. [5] used the application of black hole principle. In [16], the agents move their position under the influence of the Gbest agent (best solution obtained so far). This influential movement improves the exploitation ability of the search process. To improve the exploitation ability of GSA, Susheel et al. [7] introduced the encircle behavior of grey wolf in GSA. To provide better tradeoff between exploration and exploitation, Zhang et al. [32] used a dynamic neighbourhood-based learning strategy. In the proposed strategy, local neighbourhoods are formed randomly which further reformulated dynamically as per the guidance of population diversity. In GSA, the concept of adaptive parameter is proposed by Mirjalili et al. [13]. 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. To overcome stagnation and improve the convergence speed of GSA, Bansal et al.[2] introduced a dynamic gravitational constant which varies according to the fitness of the agents. Wang et.al. [31] proposed a three layered hierarchical GSA model having a modified gravitational constant. The proposed hierarchical model is capable to understand the population topology which further enhances the search ability of GSA. To provide a better trade-off between exploration and exploitation, Susheel et. al. [8] proposed a generic method of parameter tuning and tuned α in G. Pelusi et. al. [18] introduced hyperbolic sine functions in GSA to find the optimal value of the gravitational constant which further improves the search mechanism.

Another strong research trend towards the improvement of GSA performance is to tune the parameter α in G. For this context, a number of α-adjusting strategies have been proposed. To avoid the possibilities of premature convergence, A. Sombra et al. [26] used a fuzzy strategy to adjust the α parameter. Chaoshun Li et al. [9] introduced a hyperbolic function to model α as a variable entity with respect to iteration. This variability of α reduces the chance of premature convergence. To prevent the possibilities of premature convergence, Saeidi-Khabisi et al. [22] introduced an adaptive α strategy with the help of fuzzy logic controller. To alleviate the premature problem, Sun et al. [28] proposed a self adaptive α which is guided by the variation of an agent's position and its fitness. However, the optimal setting of α and G_0 in G is still a challenging job.

Besides the aforementioned variants of GSA, some work has been done to overcome the shortcomings of the K_{best} archive in GSA. In this context, Sun et al. [30] and Zhang et al [32] used different approaches to employ the local neighbourhood with the global topology in K_{best} archive.

In this paper, a new variant of GSA, named as 'A novel neighbourhood archives embedded gravitational constant in GSA (NAGGSA)' is proposed. The NAGGSA has the following novelties:

- To overcome the shortcomings of the K_{best} archive in the GSA model, two neighbourhood archives are proposed through which each agent obtains the best neighbours based on its current position (F archive) or its distance (D archive) from the most promising regions of the landscape. These obtained neighbours navigate the agent as per its search requirements. Additionally, the small size of these neighbourhood archives significantly reduces the computational complexity of the algorithm.
- A novel fitness-distance ratio based gravitational constant $FDG_{i,Neigh}$ is proposed which individually scales the step size of the agent X_i towards the direction of its each neighbour X_i^{Neigh} assigned by the proposed archives.

The combined effect of both proposed concepts produces a novel search mechanism that searches the optimal and sub-optimal regions, simultaneously.

The remainder of this paper is organized as follows. Section 2 briefly describes the frameworks of GSA. In Section 3, a detailed introduction of the proposed NAGGSA is given. The experimental setting and simulation results are presented in Section 4. Finally, Section 5 concludes the paper.

2 Basic Gravitational Search Algorithm

Gravitational Search Algorithm (GSA) is a new swarm intelligence technique for optimization developed by Rashedi et al [21]. 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 calculated as below:

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

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

Here, $fit_i(t)$ is the fitness value of agent X_i at iteration t and $M_i(t)$ 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 K best heavier masses in d^{th} dimension at iteration t. $F_i^d(t)$ is calculated as:

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

Here, K_{best} is an archive of first K agents with the best fitness values (say Kbest agents) and biggest masses and $rand_j$ is a uniform random number between 0 and 1. The cardinality of K_{best} archive decreases from N to 1 iteratively. At the t^{th} iteration, the force applied on agent i by agent j in the d^{th} dimension is defined as:

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

Here, $R_{ij}(t)$ is the Euclidean distance between two agents, i and j. ϵ ($\epsilon > 0$) is a small number. Finally, the acceleration of an agent in d^{th} dimension is calculated as:

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

$d = 1, 2, ..., n$ 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 Neighbourhood archives embedded gravitational constant in GSA (NAGGSA)

The performance of a meta-heuristic algorithm depends upon the social interaction among its agents. In the basic GSA, the social interaction among agents is controlled by a set of K best fit agents, named as K_{best} archive. The size of K_{best} archive is equivalent to the number of different directions an agent can have, to explore the search space in a particular evolutionary state. Figure 1 illustrates the size of K_{best} archive with respect to time (iteration). Although the K_{best} archive maintains the balance between exploration and exploitation, it has the following shortcomings:

- The K_{best} archive is common for all the agents of the swarm for a current evolutionary state. Each agent interacts with the same agents of the K_{best} archive irrespective of its individual requirement for the current search i.e., both best and worst fit agents have the same K best fit agents.
- Although the large size of K_{best} archive helps to explore the search space in the initial phase, it increases the computational complexity of the basic GSA model.
- Due to having a small size of K_{best} archive in the later iterations, GSA suffers from quick loss of search diversity.
- In the last state of the search process, the K_{best} archive has only one best fit agent (Figure 1). It means all the agents (except the best one) exploit the optimal region of the landscape but at the same time, the best fit agent does not have any agent for interaction. Therefore, in the last state of the search process, the best fit agent does not change its position.

Due to the above mentioned shortcomings in the K_{best} archive, it is replaced by two novel neighbourhood archive (D archive and F archive). Both archives provide a set of p number of neighbours for agent X_i, define as:

$$Nbd(X_i) = \{X_i^{Neigh} : Neigh = 1, ..., p\} \tag{11}$$

Insert Figure 1 here.

The following subsections describe the formulation of these proposed neighbourhood archives.

3.1 Distance based K_5 neighbourhood archive (D neighbourhood archive)

Algorithm 1 presents the pseudo-code to find the D neighbourhood archive for agent X_i. In this strategy, a set K_5 is constructed by five superior agents, say X_{K_1}, X_{K_2},...X_{K_5} from the current swarm. Let $K_5 = \{X_{K_1}, X_{K_2}, ...X_{K_5}\}$. The agents of set K_5 represent the five most promising regions of the search landscape. Out of these five promising regions, a non-K_5 agent chooses three nearest regions for the social interaction. This kind of learning enables a local approach in the global neighbourhood that provides a self diversified search as per the search requirement. On the other hand, a K_5 agent further exploits the most promising regions through the agents of its own set using the same neighbourhood mechanism. Since this neighbourhood structure emphasis nearest distance therefore if two or three regions out of five have equal nearest distances from the agent than both two or three regions will be considered as its neighbour regions. If more than three regions have equal nearest distances from the agent than the algorithm will select any three regions arbitrarily.

Algorithm 1 D neighbourhood archive:

1: Calculate the fitness of each agent of the swarm $(X_i, i = 1 : N)$;
2: Sort the fitness in ascending order;
3: Define set K_5 of five superior agents from the sorted array. $K_5 = \{X_{K_1}, X_{K_2}, ...X_{K_5}\}$;
4: Calculate $Dis_i = \{D_{iK_j} : D_{iK_j} = ||X_i, X_{K_j}||_2\ ,\ j = 1 : 5\}$
5: $Nbd(X_i) = \{X_{K_l}, X_{K_m}, X_{K_n} : where\ D_{iK_l}, D_{iK_m}, D_{iK_n}\ are\ less\ than\ other\ two\ distances\ from\ X_i\}$

3.2 Neighbourhood archive based on the agent's current fitness level (F archive)

Algorithm 2 presents a novel neighbourhood archive of each agent that is based on its current fitness. In this approach, the whole swarm is divided into five different fitness hierarchical sets (or neighbourhood classes) according to the agent's current fitness. The neighbours of an agent are decided by the set (or class) in which the agent belongs. In this neighbourhood structure, some agents of the swarm exploit the superior regions of the landscape while some agents explore the fixed regions of the landscape, simultaneously. In Algorithm 2, the agents of set S_1 always exploit the superior regions of the landscape by interacting with the other agents of the set S_1 itself. On the other hand, agents of set $S_i, (\forall i = 2 : 5)$ interact with their neighbours which belong to different level of fitness hierarchy. This kind of hierarchical model helps to avoid the possibility of stagnation. The common best agent (first agent of S_1) in each neighbourhood class navigates each agent towards the most promising region of the landscape which further accelerate the convergence speed of the algorithm. It is worth mentioning here that the best three agents of the swarm which belongs to the set S_1 are neighbours of each other. Figure 2 presents the interaction of these three agents in a landscape of the minimization problem. The best agent (red ball) posses two neighbours having opposite directions from the optimal point of the landscape. The second best agent (green ball) has two neighbours in which one is in the direction and one is away from the optimal point. The third best agent (black ball) have both neighbours in the direction of the optimal point of the landscape. By following these directions, these three agents exploit the top optimal region of the landscape in the later iterations. This kind of exploitation avoid the possibility of stagnation of the best agent in the basic GSA model which further improve the exploitation ability of the algorithm in the terminal phase of the search process.

Insert Figure 2 here.

Algorithm 2 F neighbourhood archive:

Calculate the fitness of each agent of the swarm $(X_i, i = 1 : N)$;
Sort the fitness in ascending order;
Create the five sets (S_1 *to* S_5) of agents as follows:
S_1= {Agents from 1 to $N/5$ of the sorted array};
S_2= {Agents from $(N/5 + 1)$ to $(2 \times N/5)$ of the sorted array};
S_3= {Agents from $(2 \times N/5 + 1)$ to $(3 \times N/5)$ of the sorted array};
S_4= {Agents from $(3 \times N/5 + 1)$ to $(4 \times N/5)$ of the sorted array};
S_5= {Agents from $(4 \times N/5 + 1)$ to N of the sorted array};
for i=1 to SN/5 **do**
    if X_j is the i^{th} agent of S_1 **then**
        $Nbd(X_j)$={First three agents of S_1};
    end if
    if X_j is the i^{th} agent of S_2 **then**
        $Nbd(X_j)$={First agent of S_2, first agent of S_1};
    end if
    if X_j is the i^{th} agent of S_3 **then**
        $Nbd(X_j)$={First agent of S_3, first agent of S_2, first agent of S_1};
    end if
    if X_j is the i^{th} agent of S_4 **then**
        $Nbd(X_j)$={First agent of S_4, first agent of S_3, first agent of S_2, first agent of S_1};
    end if
    if X_j is the i^{th} agent of S_5 **then**
        $Nbd(X_j)$={First agent of S_5, first agent of S_4, first agent of S_3, first agent of S_2, first agent of S_1};
    end if
end for

3.3 Selection of the neighbourhood archive

Initially, each agent X_i maintains the social interaction through F archive. Whenever the agent X_i stagnates, its neighbourhood archive is shifted from F archive to D archive. To estimate stagnation of X_i agent, a counter *ctr* is defined which counts the number of sequential iterations in which the agent does not improve itself. It is obvious that X_i is more likely to be trapped in suboptimal region as *ctr* increases. The upper limit of the counter *ctr* is set to a fixed value *stg*. If *ctr* exceeds *stg*, it means the agent X_i is facing a big risk of stagnation. Generally, the value of *stg* should be neither too large nor too small. A large *stg* value will consume more computation resources due to excessive perturbation on the agent X_i, while a small value slows the convergence speed because agents will take a long time to search around the local optimum. In this study, the value of *stg* is set to 10 (based on numerical experiments).

3.4 The proposed gravitational constant $FDG_{i,Neigh}$

In basic GSA, the gravitational constant G (equation (8)) is the exponential decreasing function of time having the following shortcomings:

- The presence of the maximum number of iterations (T) in equation (8) makes G less effective with respect to the search requirements. G scales the step size of the agent according to the pre-defined span of the search process (T), irrespective of its actual search requirements.
- The constant parameter G_0 is responsible for the initial exploration in the search process. Figure 3 shows the significance of G_0 in the evolutionary process. Different values of G_0 produce different initial exploration phases which effect the search process accordingly. Any constant value of G_0 does not provide an optimal ability to explore the search process.
- Since the reducing constant α is responsible to navigate the search process from exploration to exploitation phase. This navigation provides a good convergence speed to GSA. Figure

4 presents different navigations with respect to different values of α. Optimal navigation through a constant α is not realistic.
- The gravitational constant $G(t)$ does not have any direct relation with the search requirement. Except the current evolutionary state (t), three constant values G_0, α and T do not provide an adaptive behaviour in terms of the search requirement.
- In GSA, each agent individually interacts with the agents of K_{best} archive through the gravitational forces. This force provides a direction to the agent towards the K_{best} agent. Now the question is 'how far the agent should travel in this direction to get the maximum beneficial information about that particular region of the landscape?'. In the basic GSA, each agent travels the same distance through the common $G(t)$ towards all the directions of the common K best fit agents neglecting the fact that how beneficial it could be.

To overcome the above shortcomings of gravitational constant $G(t)$ in basic GSA, the following attributes of the proposed neighbourhood archives (either D or F archive) associated with each agent $X_i, i = 1 : N$ are used:

- The size p of X_i's neighbourhood archive
- The neighbours ($X_i^{Neigh}, Neigh = 1, ..., p$), which provides the optimal information to the agent X_i about the most promising regions of the landscape according to its necessity in terms of the current evolutionary state.
- The distance ($R_{i,Neigh}$) between the agent X_i and its neighbour X_i^{Neigh}.
- A set of fitness differences $FD_i = \{\Delta f_{i,Neigh} = f(X_i) - f(X_i^{Neigh}), Neigh = 1, ..., p\}$ which defines the position of the agent X_i related to the position of its assigned neighbours ($X_i^{Neigh}, Neigh = 1, ..., p$), in terms of the optimality. With respect to FD_i, an agent X_i can classify into three categories as follows:
Category :1 $X_i \in category\ 1 \Rightarrow \Delta f_{i,Neigh} > 0,\ \forall\ Neigh = 1, ..., p$
Category :2 $X_i \in category\ 2 \Rightarrow \Delta f_{i,Neigh} < 0,\ \forall\ Neigh = 1, ..., p$
Category :3 $X_i \in category\ 3 \Rightarrow$ some $\Delta f_{i,Neigh} > 0$ and some $\Delta f_{i,Neigh} < 0$

Insert Figure 3 here.

Insert Figure 4 here.

With the help of these attributes, a novel fitness-distance ratio based gravitational constant $FDG_{i,Neigh}(t)$ is proposed. For the current iteration t, $FDG_{i,Neigh}(t)$ individually scales the step size of the agent X_i in each direction of its neighbours $X_i^{Neigh}, Neigh = 1, ..., p$. $DFG_{i,Neigh}(t)$ is defined as:

$$FDG_{i,Neigh}(t) = MD_i + \beta \left(\frac{f_{i,Neigh}}{d_{i,Neigh}} \right)^2,\ \forall Neigh = 1, ..., p \tag{12}$$

$$MD_i = \frac{\sum_{Neigh=1}^{p} R_{i,Neigh}}{p} \tag{13}$$

$$f_{i,Neigh} = \begin{cases} \frac{\Delta f_{i,Neigh}}{\sum_{Neigh=1}^{p} \Delta f_{i,Neigh}} & \text{if } X_i \in category\ 1 \\ \frac{\delta \times |\Delta f_{i,Neigh}|}{\sum_{Neigh=1}^{p} \Delta f_{i,Neigh}} & \text{if } X_i \in category\ 2 \\ \frac{\Delta f_{i,Neigh}}{\sum_{Neigh=1}^{p} \Delta f_{i,Neigh}} & \text{if } X_i \in category\ 3 \text{ and } \Delta f_{i,Neigh} > 0 \\ \frac{\gamma \times |\Delta f_{i,Neigh}|}{\sum_{Neigh=1}^{p} \Delta f_{i,Neigh}} & \text{if} X_i \in category\ 3 \text{ and } \Delta f_{i,Neigh} < 0 \end{cases} \tag{14}$$

$$d_{i,Neigh} = \frac{R_{i,Neigh}}{\sum_{Neigh=1}^{p} R_{i,Neigh}}, \ \forall \ Neigh = 1, ..., p \tag{15}$$

where MD_i is the mean distance of the agent X_i with its current neighbours. $d_{i,Neigh}$ and $f_{i,Neigh}$ are the normalized distance and normalized fitness difference between the agent X_i and its neighbour X_i^{Neigh}, respectively. β is a linearly decreasing function from 1 to 0 over the course of iterations. It is clear from equation (12) that $FDG_{i,Neigh} \propto f_{i,Neigh}$. It means that the step size of the agent X_i in the direction of its individual neighbour X_i^{Neigh} is directly proportional to the fitness difference ($\Delta f_{i,Neigh}$) between them.

Figure 5 presents all the above three categories for the minimization problem. subgraph (5(a)), subgraph (5(b)) and subgraph (5(c)) present category 1, category 2 and category 3, respectively. In the subgraph (5(a)), all the neighbours of agent X_i belong to the better optimal regions of the landscape compare to its current position. The social interaction by these neighbours navigates the agent towards the more optimal regions of the landscape which further increases the convergence rate of the search process. On the contrary, in Figure (5(b)) the agent itself has the optimal position compare to all its neighbours. Although all the less fit neighbours downgrade the fitness of the agent, this kind of social interaction avoids the stagnation possibility on the best fit agent (either the global best agent or the currently best agent of the swarm). The red ball in Figure 2 presents the mentioned state of the best fit agent for the minimization problem. Moreover, these interactions with a large step size may lead the best fit agent far away from the optimal region of the landscape, against the search requirement. Therefore, the small step sizes are beneficial for making this category as a stagnation avoidance tool for the multi-modal landscape. To do so, a small positive value δ is used to reduce the step size for category 2 in the equation (14). Finally, Figure (5(c)) presents the third category which is responsible to maintain the diversity of the search due to having both less and more fit neighbours compare to the agent itself. Like category 2, the agent should have a small step size to interact with its less fit neighbour. In this regards, a small positive value $\gamma < \delta$ is used for $\Delta f_{i,Neigh} < 0$ in category 3. Further, the ratio ($\frac{f_{i,Neigh}}{d_{i,Neigh}}$) in equation (12) controls the step size of the agent more precisely. This ratio provides the maximum weight to the nearest best fit neighbour for the agent's next move. Due to its monotonic decreasing behaviour, β annihilates the effect of $d_{i,Neigh}$ in the terminal phase of the search process.

Under the influence of the proposed $FDG_{i,Neigh}$, the agent X_i interacts with its neighbour X_i^{Neigh} through the gravitational force defined as

$$F_{i,Neigh}(t) = FDG_{i,Neigh} \frac{M_i(t) M_{Neigh}(t)}{R_{i,Neigh} + \epsilon} (X_i(t) - X_i^{Neigh}(t)) \tag{16}$$

Finally, the total gravitational force acting on the agent X_i by its all neighbours of its neighbourhood archive for the current iteration (t) is defined as:

$$F_i(t) = \sum_{Neigh=1}^{p} F_{i,Neigh}(t) \tag{17}$$

Insert Figure 5 (subfigures as Figure5a, Figure5b and Figure5c) here.

Insert Figure 6 (subfigures as Figure6a, Figure6b, Figure6c and Figure6d) here.

Insert Figure 7 (subfigures as Figure7a, Figure7b, Figure7c and Figure7d) here.

Insert Figure 8 (subfigures as Figure8a, Figure8b, Figure8c and Figure8d) here.

Insert Figure 9 (subfigures as Figure9a, Figure9b, Figure9c and Figure9d) here.

Algorithm 3 NAGGSA algorithm for minimization problem:

/*Initialization*/
Initialize the position and velocities of agents
ctr=0, stg=10;
/*Main Loop*/
while Termination criteria is not satisfied **do**
    Calculate the fitness of each agent of the swarm
    Find best and worst fitness
    Calculate masses of each agents
    for i=1 to SN **do**
        if Iteration>1 **then**
            if fitness(X_i) $<$ fitness old(X_i) **then**
                ctr=0;
            else
                ctr=ctr+1;
            end if
        end if
        if $ctr < stg$ **then**
            X_i follows F neighbourhood archive (using Algorithm 2)
        else
            X_i follows D neighbourhood archive (Algorithm 1)
        end if
        With respect to the selected neighbourhood archive, calculate the proposed $FDG_{i,Neigh}$ (using equation (12)) of X_i for its neighbours
        Calculate total force (using equation (17)) acting on X_i by its neighbours
        Find acceleration for X_i
    end for
    fitness old =fitness;
    Update velocities and positions of agents
end while

The implementation of the proposed NAGGSA is summarized in Algorithm 3. Figures 6, 7, 8 and 9 present different attributes of the first agent X_1 of the swarm under the mechanism of the proposed NAGGSA on f_1 (Unimodal function), f_5 (Multimodal function), f_8 (Hybrid function) and f_{11} (Composite function) (refer section 4.1). In each figure, the first subgraph (a) presents the number of neighbours of the first agent X_1 of the swarm through out the search process provided by either F or D archive. It is clear from Subgraph (a) that the agent can possess minimum 2 and maximum 5 neighbours in any evolutionary state. Subgraph (b) presents the mean distance of X_1 with its neighbours. Subgraph (c) presents the proposed $FDG_{1,Neigh}(t)$ of X_1 for its neighbours associated with the basic gravitational constant $G(t)$ of GSA model. For the better graphical analysis, a magnified version of subgraph (c) is presented in subgraph (d).

4 Results and Discussion

4.1 Testbeds under consideration

In this section, the proposed NAGGSA is tested over 12 unconstrained continuous test functions of CEC 2015 test suite (Testbed 1)[12] and 22 test functions of CEC 2014 test suite (Testbed 2)

[11]. According to the different topological characteristics, the 12 test functions of Testbed 1 are categorized into four groups : uni-modal functions ($f_1(F1_{cec15})$), simple multi-modal functions ($f_2(F3_{cec15})$ and $f_3(F4_{cec15})$), hybrid functions ($f_4(F6_{cec15})$, $f_5(F7_{cec15})$ and $f_6(F8_{cec15})$) and composite functions ($f_7(F9_{cec15})$, $f_8(F10_{cec15})$, $f_9(F12_{cec15})$, $f_{10}(F13_{cec15})$, $f_{11}(F14_{cec15})$ and $f_{12}(F15_{cec15})$). The 22 test functions of Testbed 2 are categorized into three groups: uni-modal functions ($g_1(F1_{cec14})$-$g_3(F3_{cec14})$), simple multi-modal functions ($g_4(F4_{cec14})$-$g_{16}(F16_{cec14})$) and hybrid functions ($g_{17}(F17_{cec14})$-$g_{22}(F22_{cec14})$). These 34 benchmark functions involve a diverse set of characteristics namely, multimodality, impurity, ill-conditioning and rotation, which can be utilized to completely investigate the optimization performance of the NAGGSA algorithm. The dimension (n) and the range of the search space of both testbeds are 30 and $[-100, 100]$, respectively.

4.2 Experimental setting

In order to validate the effectiveness and robustness of proposed algorithm, NAGGSA is compared with basic GSA along with some state-of-the-art algorithms like Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [6], Biogeography-based optimization (BBO) [25], Disruption in biogeography-based optimization (DBBO) [1], Differential evolution (DE) [27] and Grey wolf optimizer (GWO)[17]. NAGGSA is also compared with some recent variants of GSA like MGSA[9], Fuzzy gravitational search algorithm (FGSA) [22], $FS\alpha$(Increase) [26], $FS\alpha$(Decrement) [26] and SCAA [28]. All the comparisons have been done over Testbed 1 with the popular experimental setting (as per recommendations of CEC 2015 test suite) given as follows:

4.2.1 Experimental setting for Test bed 1

- The number of simulations/run =51,
- Swarm size N=50,
- Dimension n=30,
- The maximum number of function evaluations for the stopping criteria of the algorithms are set to be $10,000 \times n$,
- Parameters for the algorithms GSA [21], CMA-ES [6], BBO [25], DBBO [1], DE [27] and GWO [17] are considered from the corresponding resources while the results of all considered GSA variants are reproduced from SCAA [28].

For further evaluation of NAGGSA, it is tested over testbed 2 along with the original GSA [21] and four competitive GSA variants namely, GAGSA [29], PSOGSA [15], FVGGSA [2] and PTGSA [8]. Since the results of GSA [21], GAGSA [29] and PSOGSA [15] over Testbed 2 are reproduced from [30]. Therefore for fair comparison, the experimental setting for Testbed 2 has been adopted from [30] and given as follows:

4.2.2 Experimental setting for Test bed 2

- The number of simulations/run =30,
- Swarm size N=60,
- Dimension n=30,
- The maximum number of function evaluations for the stopping criteria of the algorithms are set to be $60,000$,
- Parameters for the algorithms GSA [21], GAGSA [29], PSOGSA [15], FVGGSA [2] and PTGSA [8] are considered from the corresponding resources.

4.3 Result and statistical analysis of experiments

4.3.1 Testbed 1

Following the experimental setting explained in section 4.2.1, the searching behavior of the proposed NAGGSA is compared with some state-of-the-art algorithms over Testbed 1. The experimental results of fitness errors are summarized in Table 1. Table 1 lists the three metrics of fitness error: Mean error(Mean), Standard Deviation of error (SD) and Wilcoxon Signed-Rank Test (h-value) [4]. The fitness error is the absolute difference between the best fitness value obtained by the algorithm and the actual global optimum of the optimization problem. The Mean and SD of these results validate the searching accuracy of the algorithms, while Wilcoxon Signed-Rank Test checks whether the results obtained by NAGGSA and other considered algorithms are significantly different or not. This non-parametric statistical test is performed on these results at 5% level of significance with the null hypothesis, 'There is no significant difference between the results' obtained by NAGGSA and other considered algorithms. In Table 1, '+' or '-' h-value indicates that NAGGSA is significantly better or worse than the other considered algorithms, while '=' h-value stands for similar performance between NAGGSA and others. The bold entries indicate the best results. As shown in Table 1, NAGGSA outperforms others in terms of mean value for 7 test functions including one unimodal (f_1), one multi-modal(f_2), two hybrid (f_4 and f_6) and three composite functions (f_8, f_9 and f_{12}). Among all metrics of comparison NAGGSA proves its supremacy over 5 test functions (f_1, f_4, f_6, f_8 and f_9). Furthermore, 43 '+' h-value out of 60 comparison confirms that the proposed NAGGSA is a significantly better algorithm than other considered algorithms. To further verify the exploitation of NAGGSA, the convergence behavior of the considered algorithms over unimodal (f_1), hybrid (f_4) and composite (f_8) functions is illustrated in Figure 10. It is clear from Figure 10 that NAGGSA outperforms others in terms of exploitation ability due to its fastest convergence rate.

Insert Table 1 here

In order to show the efficiency of the proposed NAGGSA over the recent variants of GSA, five GSA variants namely, MGSA-α [9], Fuzzy GSA [22], $FS\alpha$(Increase) [26], $FS\alpha$(Decrement) [26] and SCAA [28] are considered for comparison. Table 2 presents the experimental results of fitness errors with two metrics: Mean and SD. The bold entries indicate the best results. Except result of NAGGSA, other results are reproduced from SCAA [28]. As per the results shown in Table 2, in terms of the mean value, NAGGSA have the better search accuracy for eight test functions (f_3, f_4, f_6, f_8, f_{10} and f_{12}). For f_1, NAGGSA is better than others except SCAA. For f_7, NAGGSA is better than Fuzzy GSA, FSα(Increase) and FSα(Decrement). For five test functions (f_4, f_6, f_8, f_{10} and f_{12}) NAGGSA outperforms others in both metrics of comparison. f_2, f_7, f_9 and f_{11} are the problems for which NAGGSA is not better in either criteria. However only for f_2, FSα (Decrement) is better than NAGGSA. While for f_5, MGSA-α is better than NAGGSA. For other problems no single algorithm is better than NAGGSA in both criteria. Therefore, overall NAGGSA works better than all other α-adjusting variants of GSA.

Based on the comparison of NAGGSA with state-of-the-art algorithms and the recent variants of GSA, it is clear that NAGGSA have an excellent search mechanism for unimodal, multi-modal, hybrid and composite test functions.

Insert Table 2 here.

Insert Figure 10 (subfigures as Figure10a, Figure10b and Figure10c) here.

4.3.2 Testbed 2

In order to show the efficiency of NAGGSA more clearly, it is re-evaluated over 22 benchmark functions of Testbed 2. Table 3 presents the experimental results which are followed by the the experimental setting given in section 4.2.2. The criteria of comparison are mean (Mean), best (Best) and standard deviation (SD) of the error values. The bold entries indicate the best results. Table 3 shows that NAGGSA outperforms in terms of mean value for the functions g_1, g_2, g_5, g_9, g_{11}, g_{14}, g_{16}, g_{20}, g_{21} and g_{22}. For g_3, g_7 and g_{17}, NAGGSA is better than others except PTGSA. For g_8, NAGGSA is better than others except PSOGSA. For g_{10}, NAGGSA is better than GAGSA and PSOGSA. For g_{12} and g_{15}, NAGGSA is better than GSA and GAGSA. For g_{18}, NAGGSA is better than GAGSA, PSOGSA and PTGSA. For g_4 and g_{13}, NAGGSA is better than GAGSA, PSOGSA and FVGGSA. For g_{19}, NAGGSA is better than others except GSA. For g_6, NAGGSA is better than GAGSA only. In terms of the best value, NAGGSA outperforms for the functions g_1, g_2, g_3, g_4, g_5, g_8, g_9, g_{10}, g_{11}, g_{14}, g_{16}, g_{17}, g_{18}, g_{20}, g_{21} and g_{22}. For three functions (g_2, g_{20} and g_{22}), NAGGSA outperforms others in all criteria of comparison. Based on these results, it is clear that the proposed NAGGSA performs significantly well for unimodal, multi-modal and hybrid test problems under shifted and rotated conditions.

Insert Table 3 here.

To statistically compare the performance of all the above algorithms simultaneously, a two-stage method (i.e., the statistical Friedman test and then a post-hoc test) is used. This non-parametric statistical test is performed pairwise at 1% level of significance with the null hypothesis, 'There is no significant difference between the results obtained by the considered pair'. The Friedman test p-value is $2.382E - 12$, that clearly indicates the significant difference between the performance of the algorithms even at 1% level of significance. According to Friedman test results, a post-hoc statistical analysis is needed. In this study, for pairwise comparisons, we also reported the adjusted p-values achieved by five post-hoc test procedures. All these procedures are implemented in R [20, 19]. Table 4 presents the p-values for each comparison which involves the proposed algorithm. From Table 4, the following observations are made:

- For all considered post-hoc test procedures, the proposed NAGGSA is significantly better than other considered algorithms.
- Based on the multiple comparison analysis, the proposed NAGGSA is an overall better algorithm as compare to other considered GSA variants.

Insert Table 4 here.

5 Conclusion

In this paper, two novel neighbourhood archive (D archive and F archive) are proposed to overcome the shortcomings of the K_{best} archive in basic GSA model. In D archive, each agent extracts the information of its nearest most promising regions of the landscape. This kind of learning enables the algorithm to sufficiently explore the feasible search space. On the other hand, the F archive provides a five level of neighbourhood strategy based on the agent's current fitness. The first level neighbourhood avoids the possibility of global best stagnation and exploits the most promising region of the landscape while second to fifth level neighbourhood helps to explore the different regions of the landscape. Since these small size archives are accountable for the social interaction therefore the proposed variant reduces the computational complexity compared to the K_{best} archive of GSA model. Secondly, a novel fitness-distance based gravitational constant is proposed which scales the agent's next move in each direction of its neighbours. The performance of the proposed variant is compared with some state-of-the-art algorithms along with some recent variants of GSA over CEC 2015 and CEC 2014 test suites. Based on the comparisons, NAGGSA has proved its excellent search ability for shifted unimodal, shifted and rotated multimodal, hybrid and composite test problems. The efficiency of the proposed variant is based on the fitness distance ratio criteria which employs a local search tool in each global search of an individual agent. This embedded local search tool can be improve the performance of the proposed variant in the binary search space, more efficiently. Therefore, In future, the proposed variant can be applied on combinatorial optimization problems of binary and discrete search space.

Declaration

Funding: The first author acknowledges the funding from South Asian University New Delhi, India and the last author acknowledges the funding from Liverpool Hope University, UK to carry out this research.
Conflicts of interest/Competing interests: All the authors declare that they has no conflict of interest.
Availability of data and material: Not applicable.
Code availability: Not applicable.

References

1. 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.
2. Jagdish Chand Bansal, Susheel Kumar Joshi, and Atulya K. Nagar. Fitness varying gravitational constant in gsa. *Applied Intelligence*, 2018.
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. Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.
5. 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.
6. Nikolaus Hansen. The cma evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.
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. Susheel Kumar Joshi and Jagdish Chand Bansal. Parameter tuning for meta-heuristics. *Knowledge-Based Systems*, 189:105094, 2020.
9. Chaoshun Li, Hongshun Li, and Pangao Kou. Piecewise function based gravitational search algorithm and its application on parameter identification of avr system. *Neurocomputing*, 124:139 – 148, 2014.

10. Xiangtao Li, Minghao Yin, and Zhiqiang Ma. Hybrid differential evolution and gravitation search algorithm for unconstrained optimization. *International Journal of Physical Sciences*, 6(25):5961–5981, 2011.
11. JJ Liang, BY Qu, and PN Suganthan. Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2013.
12. JJ Liang, BY Qu, PN Suganthan, and Q Chen. Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization. *Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2014.
13. Seyedali Mirjalili and Amir H Gandomi. Chaotic gravitational constants for the gravitational search algorithm. *Applied Soft Computing*, 2017.
14. 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.
15. SeyedAli Mirjalili, Siti Zaiton Mohd Hashim, and Hossein Moradian Sardroudi. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, 218(22):11125–11137, 2012.
16. Seyedali Mirjalili and Andrew Lewis. Adaptive gbest-guided gravitational search algorithm. *Neural Computing and Applications*, 25(7-8):1569–1584, 2014.
17. Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.
18. Danilo Pelusi, Raffaele Mascella, Luca Tallini, Janmenjoy Nayak, Bighnaraj Naik, and Yong Deng. Improving exploration and exploitation via a hyperbolic gravitational search algorithm. *Knowledge-Based Systems*, 193:105404, 2020.
19. Thorsten Pohlert. *The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR)*, 2014. R package.
20. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.
21. Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.
22. F. 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.
23. S Sarafrazi, H Nezamabadi-Pour, and S Saryazdi. Disruption: a new operator in gravitational search algorithm. *Scientia Iranica*, 18(3):539–548, 2011.
24. 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.
25. Dan Simon. Biogeography-based optimization. *Evolutionary Computation, IEEE Transactions on*, 12(6):702–713, 2008.
26. A. Sombra, F. Valdez, P. Melin, and O. Castillo. A new gravitational search algorithm using fuzzy logic to parameter adaptation. In *2013 IEEE Congress on Evolutionary Computation*, pages 1068–1074, June 2013.
27. Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

28. Genyun Sun, Ping Ma, Jinchang Ren, Aizhu Zhang, and Xiuping Jia. A stability constrained adaptive alpha for gravitational search algorithm. *Knowledge-Based Systems*, 139:200–213, 2018.
29. Genyun Sun and Aizhu Zhang. A hybrid genetic algorithm and gravitational search algorithm for image segmentation using multilevel thresholding. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 707–714. Springer, 2013.
30. Genyun Sun, Aizhu Zhang, Zhenjie Wang, Yanjuan Yao, Jinsheng Ma, and Gary Douglas Couples. Locally informed gravitational search algorithm. *Knowledge-Based Systems*, 104:134–144, 2016.
31. Yirui Wang, Yang Yu, Shangce Gao, Haiyu Pan, and Gang Yang. A hierarchical gravitational search algorithm with an effective gravitational constant. *Swarm and Evolutionary Computation*, 46:118–139, 2019.
32. Aizhu Zhang, Genyun Sun, Jinchang Ren, Xiaodong Li, Zhenjie Wang, and Xiuping Jia. A dynamic neighborhood learning-based gravitational search algorithm. *IEEE transactions on cybernetics*, 2016.

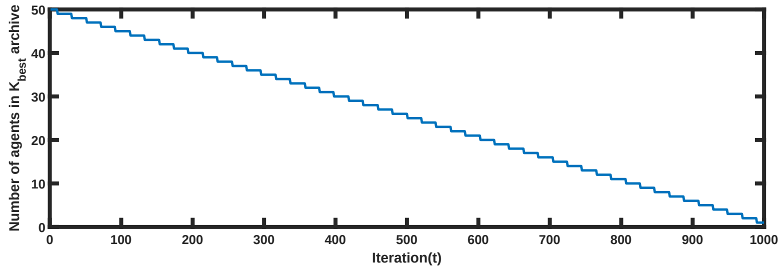


Fig. 1 Number of agents in K_{best} archive

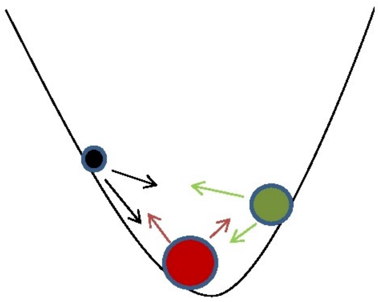


Fig. 2 The three best agents of S_1 set

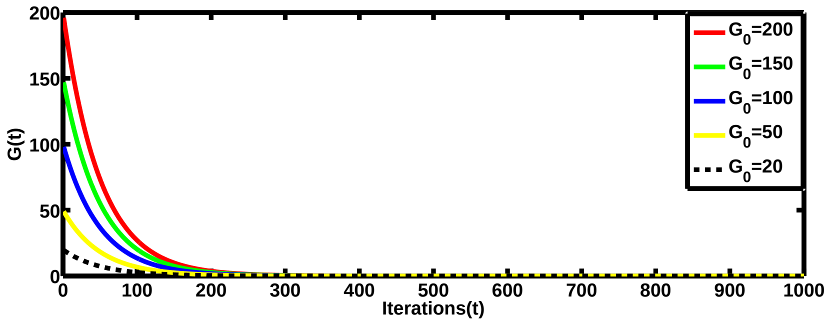


Fig. 3 Different $G(t)$ with respect to different G_0 values (α is fixed to 20)

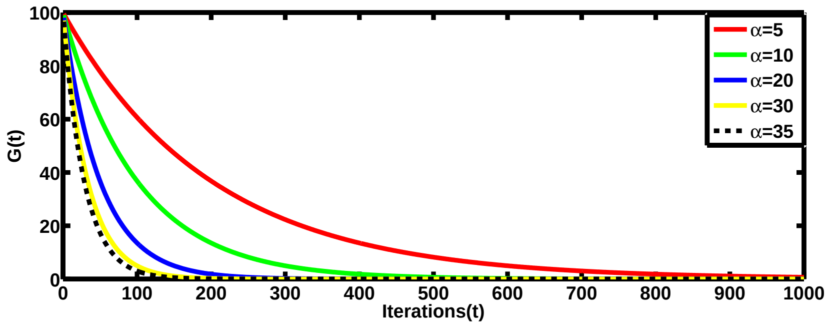
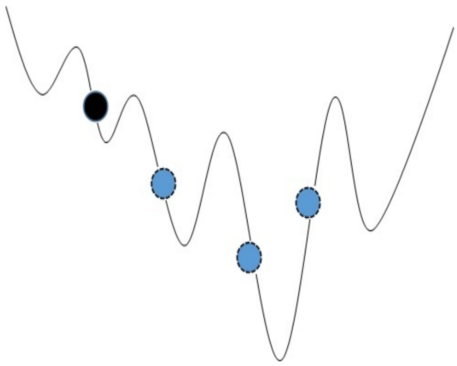
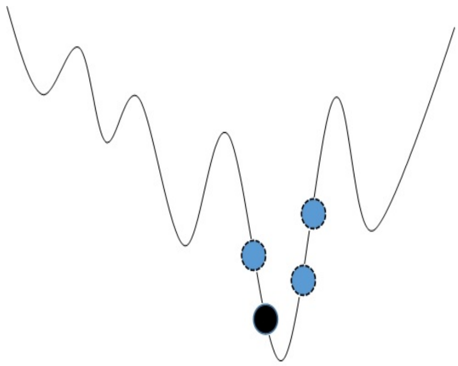


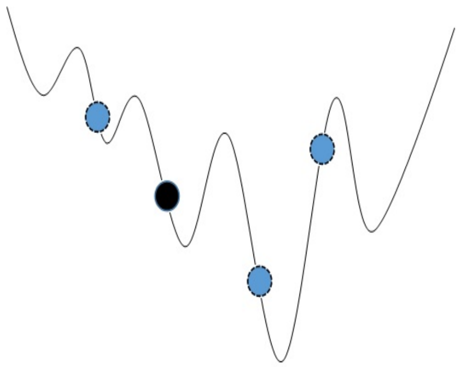
Fig. 4 $G(t)$ with respect to different α values (G_0 is fixed to 100)



(a) category 1

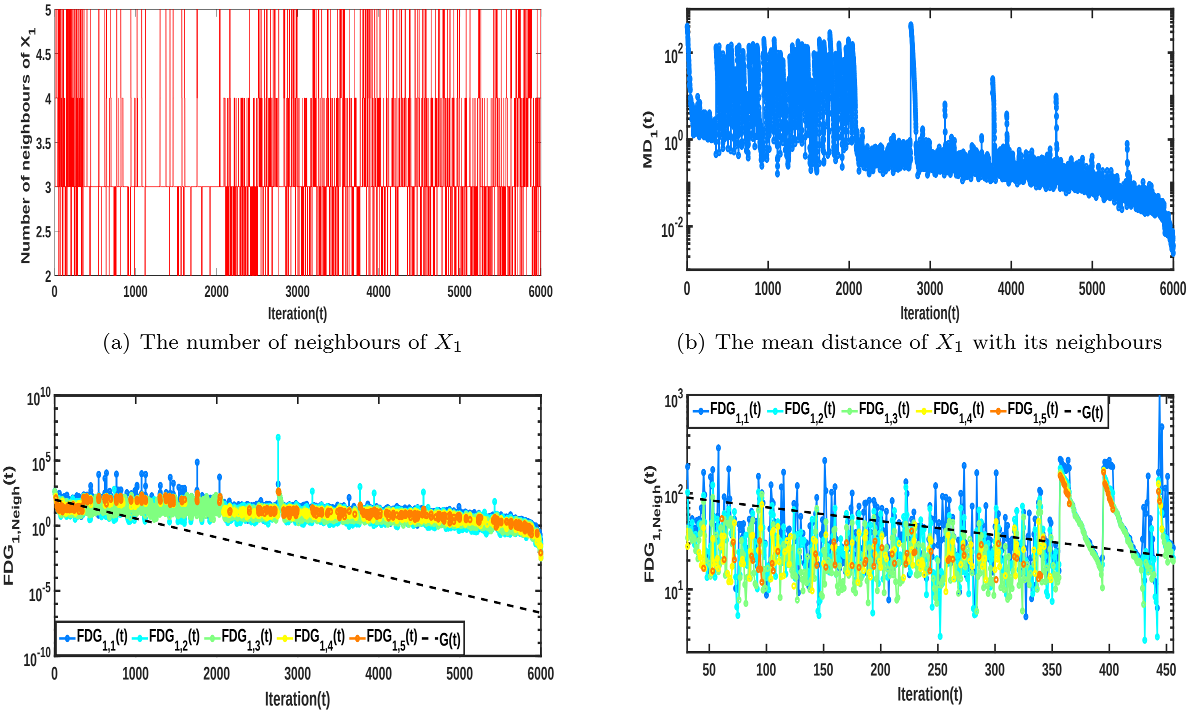


(b) category 2



(c) category 3

Fig. 5 The 2-D graphical representation of the three categories with respect to the position of an agent X_i (black ball) and its neighbours (blue balls) for the minimization problem



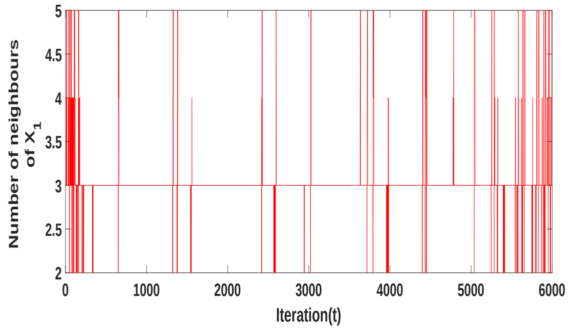
(a) The number of neighbours of X_1

(b) The mean distance of X_1 with its neighbours

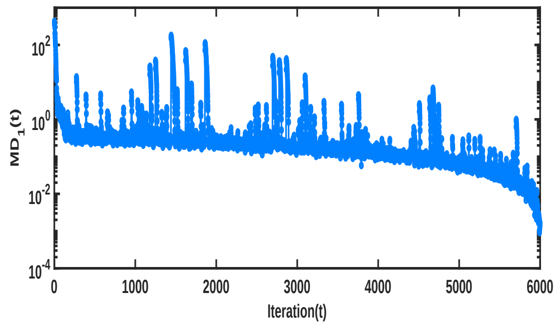
(c) The proposed $FDG_{1,Neigh}(t)$ with basic $G(t)$

(d) Magnified version of figure (c)

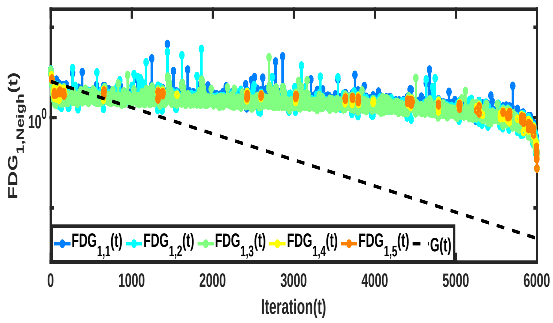
Fig. 6 The different attributes of the first agent X_1 for f_1 (uni-modal function)



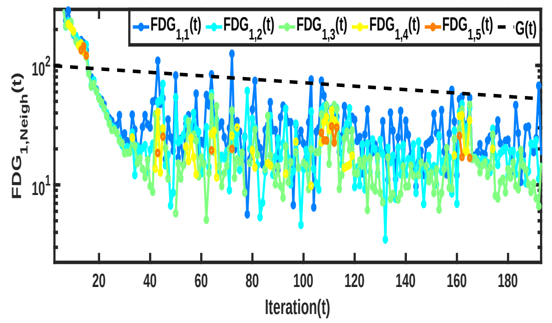
(a) The number of neighbours of X_1



(b) The mean distance of X_1 with its neighbours

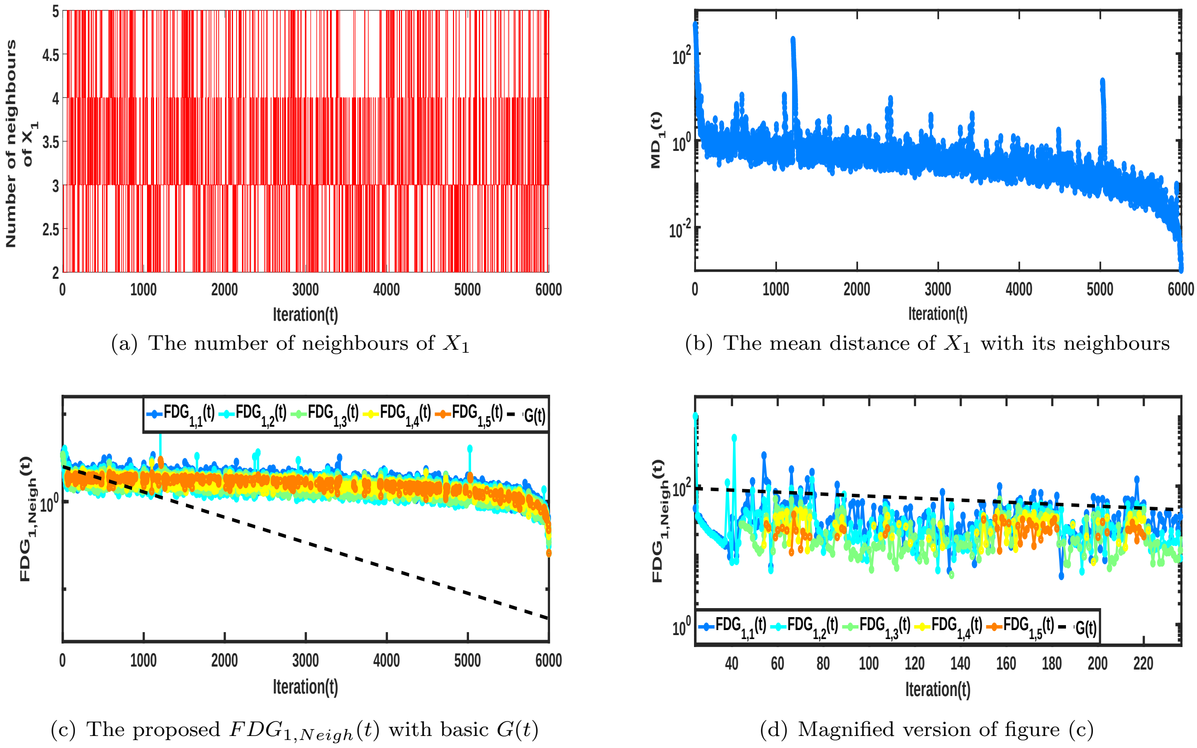


(c) The proposed $FDG_{1,Neigh}(t)$ with basic $G(t)$



(d) Magnified version of figure (c)

Fig. 7 The different attributes of the first agent X_1 for f_5 (Multimodal function)



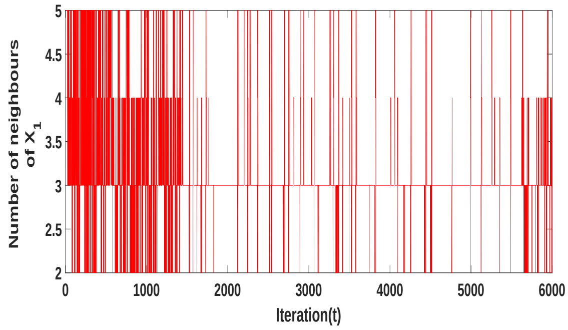
(a) The number of neighbours of X_1

(b) The mean distance of X_1 with its neighbours

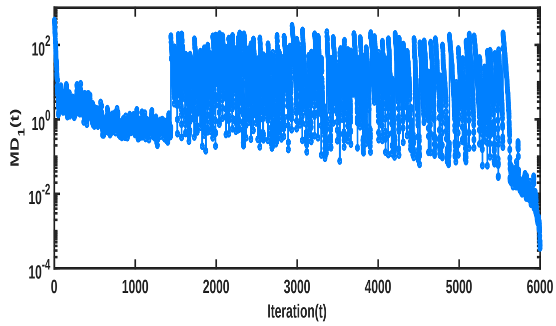
(c) The proposed $FDG_{1,Neigh}(t)$ with basic $G(t)$

(d) Magnified version of figure (c)

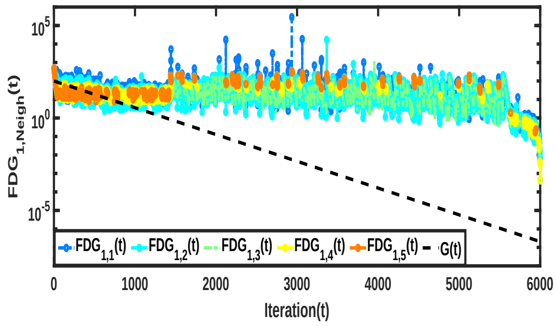
Fig. 8 The different attributes of the first agent X_1 for f_8 (Hybrid function)



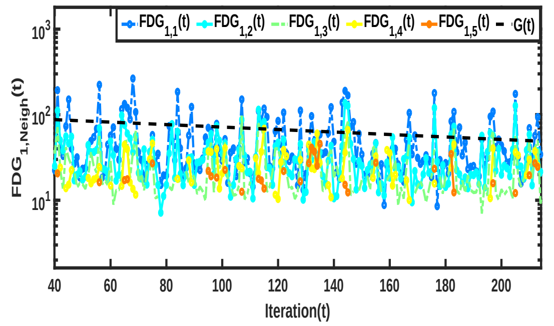
(a) The number of neighbours of X_1



(b) The mean distance of X_1 with its neighbours

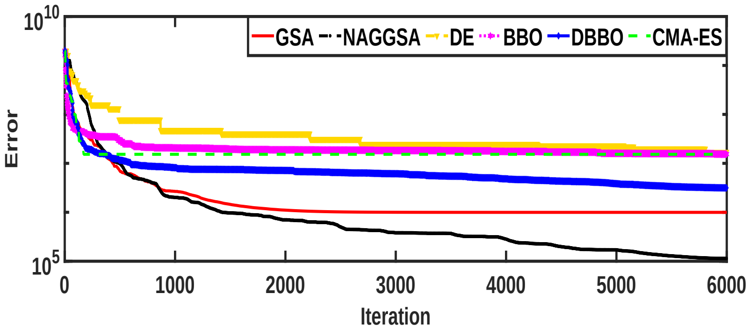


(c) The proposed $FDG_{1,Neigh}(t)$ with basic $G(t)$

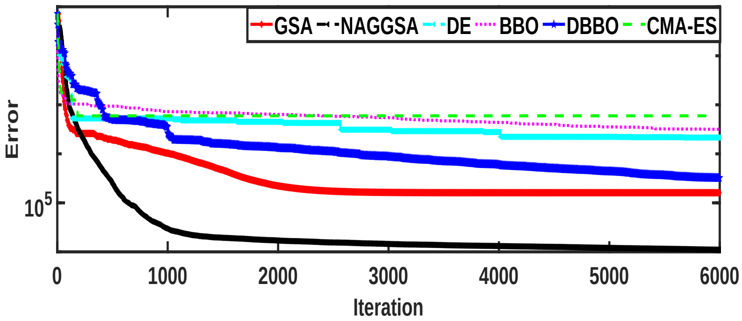


(d) Magnified version of figure (c)

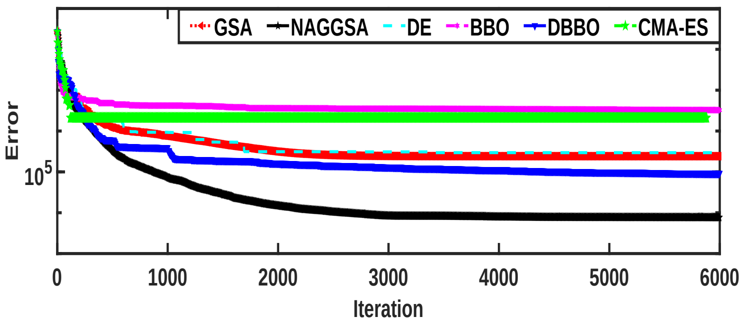
Fig. 9 The different attributes of the first agent X_1 for f_{11} (Composition function)



(a) Convergence performance comparison for minimizing of unimodal function f_1



(b) Convergence performance comparison for minimizing of hybrid function f_4



(c) Convergence performance comparison for minimizing of composition function f_8

Fig. 10 Convergence graphs for function f_1, f_4 and f_8

Table 1: Fitness errors of NAGGSA along with the considered state-of-the-art algorithms over Testbed 1 (**TP** denotes Test Problem under consideration and **SD** stands for Standard Deviation)

TP	**metrics**	**GSA**	**CMA-ES**	**BBO**	**DBBO**	**DE**	**GWO**	**NAGGSA**
$f_1(F1_{cec15})$	Mean	8.12E+05	2.04E+07	8.63E+06	3.77E+06	1.52E+07	2.01E+07	**4.32E+05**
	SD	3.42E+05	7.39E+06	4.51E+06	2.31E+06	3.14E+06	2.41E+07	**1.23E+06**
	h value	(+)	(+)	(+)	(+)	(+)	(+)	
$f_2(F3_{cec15})$	Mean	**2.00E+01**	2.10E+01	2.01E+01	**2.00E+01**	2.07E+01	2.09E+01	**2.00E+01**
	SD	6.80E-05	5.22E-02	2.96E-02	**1.44E-07**	5.20E-02	5.10E-02	7.79E-05
	h value	(-)	(+)	(+)	(=)	(+)	(+)	
$f_3(F4_{cec15})$	Mean	2.12E+02	1.16E+02	**6.09E+01**	8.55E+01	1.20E+02	9.21E+01	1.70E+02
	SD	1.93E+01	6.62E+01	1.36E+01	2.31E+01	**1.06E+01**	3.51E+01	2.56E+01
	h value	(+)	(-)	(-)	(-)	(-)		
$f_4(F6_{cec15})$	Mean	1.33E+05	2.64E+06	4.14E+06	9.93E+05	1.47E+06	1.16E+06	**7.72E+03**
	SD	5.22E+04	1.46E+06	3.10E+06	1.01E+06	7.05E+05	9.46E+05	**5.57E+03**
	h value	(+)	(+)	(+)	(+)	(+)	(+)	
$f_5(F7_{cec15})$	Mean	1.54E+01	**8.67E+00**	1.47E+01	1.72E+01	1.27E+01	1.87E+01	2.11E+01
	SD	9.09E+00	9.11E-01	1.32E+01	1.93E+01	**6.27E-01**	3.36E+00	4.74E+00
	h value	(-)	(-)	(-)	(=)	(+)		
$f_6(F8_{cec15})$	Mean	2.41E+04	1.90E+06	2.12E+06	3.30E+05	2.87E+05	2.78E+05	**1.17E+04**
	SD	9.84E+03	1.22E+06	2.13E+06	5.69E+05	1.10E+05	3.97E+05	**5.62E+03**
	h value	(+)	(+)	(+)	(+)	(+)	(+)	
$f_7(F9_{cec15})$	Mean	1.37E+02	1.52E+02	1.05E+02	**1.03E+02**	**1.03E+02**	1.12E+02	1.67E+02
	SD	1.02E+02	9.38E+01	6.08E-01	2.47E-01	**1.88E-01**	2.33E+02	1.59E+02
	h value	(-)	(+)	(+)	(-)	(-)	(=)	
$f_8(F10_{cec15})$	Mean	3.98E+05	2.54E+06	2.04E+06	2.85E+05	4.66E+05	1.41E+06	**3.43E+04**
	SD	1.49E+05	1.58E+06	1.50E+06	8.33E+05	1.94E+05	1.05E+06	**7.79E+04**
	h value	(+)	(+)	(+)	(+)	(+)	(+)	
$f_9(F12_{cec15})$	Mean	1.04E+02	1.92E+02	1.09E+02	1.85E+02	1.07E+02	1.10E+02	**1.01E+02**
	SD	8.45E-01	2.56E+01	1.59E+00	3.38E+01	6.24E-01	1.84E+01	**5.93E-01**
	h value	(+)	(+)	(+)	(+)	(+)	(+)	

Table 1 Continued:

TP	metrics	GSA	CMA-ES	BBO	DBBO	DE	GWO	NAGGSA
$f_{10}(F13_{cec15})$	Mean	1.38E+03	6.95E-03	3.58E-02	**6.14E-03**	2.59E-02	5.36E-02	1.05E+02
	SD	1.26E+03	**9.72E-05**	4.00E-03	2.48E-04	2.22E-04	2.23E-02	8.79E+01
	h value	(+)	(-)	(-)	(-)	(-)		
$f_{11}(F14_{cec15})$	Mean	**1.00E+02**	1.02E+04	3.34E+04	6.75E+03	3.35E+04	3.58E+04	3.20E+04
	SD	**9.63661E-08**	7.72E+03	1.05E+03	9.55E+03	2.96E+02	2.39E+03	1.87E+04
	h value	(-)	(+)	(+)	(-)	(+)	(+)	
$f_{12}(F15_{cec15})$	Mean	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	1.27E+02	**1.00E+02**
	SD	1.34832E-10	1.41E-13	2.98E-02	1.12E-03	**1.25E-13**	1.79E+01	4.45E-05
	h value	(=)	(=)	(+)	(+)	(=)	(+)	

Table 2: Fitness errors of NAGGSA along with α adjusting GSA variants over Testbed 1 (**TP** denotes Test Problem under consideration and **SD** stands for Standard Deviation)

TP	**metrics**	**MGSA-α**	**FuzzyGSA**	**FSα(Increase)**	**FSα(Decrement)**	**SCAA**	**NAGGSA**
$f_1(F1_{cec15})$	Mean	1.012E+06	2.707E+06	2.046E+06	1.428E+07	**4.093E+05**	4.32E+05
	SD	3.907E+05	5.063E+06	1.297E+06	8.813E+06	**2.405E+05**	1.23E+06
$f_2(F3_{cec15})$	Mean	**2.000E+01**	**2.000E+01**	**2.000E+01**	**2.000E+01**	2.094E+01	**2.000E+01**
	SD	6.594E-05	8.113E-05	1.109E-04	**6.150E-05**	5.840E-02	7.79E-05
$f_3(F4_{cec15})$	Mean	1.963E+02	2.194E+02	2.084E+02	2.363E+02	2.173E+02	**1.70E+02**
	SD	2.811E+01	1.924E+01	2.023E+01	2.261E+01	2.223E+01	**2.56E+01**
$f_4(F6_{cec15})$	Mean	3.553E+05	6.856E+05	9.472E+05	1.704E+06	5.587E+04	**7.72E+03**
	SD	1.725E+05	2.962E+05	3.593E+05	6.204E+05	2.566E+04	**5.57E+03**
$f_5(F7_{cec15})$	Mean	1.524E+01	2.236E+01	2.441E+01	6.397E+01	**9.779E+00**	2.11E+01
	SD	9.643E+00	1.949E+01	2.058E+01	2.388E+01	**3.133E+00**	4.74E+00
$f_6(F8_{cec15})$	Mean	2.388E+04	3.050E+04	5.557E+04	1.007E+05	2.154E+04	**1.17E+04**
	SD	7.509E+03	1.152E+04	3.344E+04	1.141E+05	8.329E+03	**5.62E+03**
$f_7(F9_{cec15})$	Mean	1.265E+02	**1.151E+02**	1.262E+02	2.025E+02	1.358E+02	1.67E+02
	SD	8.221E+01	1.218E+02	**7.943E+01**	1.627E+02	1.016E+02	1.59E+02
$f_8(F10_{cec15})$	Mean	6.936E+05	9.961E+05	1.280E+06	2.485E+06	1.921E+05	**3.43E+04**
	SD	2.310E+05	3.994E+05	6.108E+05	1.004E+06	5.998E+04	**7.79E+04**
$f_9(F12_{cec15})$	Mean	1.036E+02	1.053E+02	1.047E+02	1.449E+02	1.034E+02	**1.020E+02**
	SD	8.215E-01	1.104E+00	9.304E-01	2.771E+01	7.031E-01	**5.95E-01**
$f_{10}(F13_{cec15})$	Mean	4.759E+03	1.673E+03	1.602E+03	2.100E+03	1.550E+03	**1.05E+02**
	SD	3.987E+03	**1.083E+03**	1.571E+03	1.121E+03	1.296E+03	8.79E+01
$f_{11}(F14_{cec15})$	Mean	**1.000E+02**	**1.000E+02**	**1.000E+02**	2.821E+04	**1.000E+02**	3.20E+04
	SD	8.716E-13	6.668E-10	0.000E+00	7.551E+03	**3.565E-13**	1.87E+04
$f_{12}(F15_{cec15})$	Mean	**1.000E+02**	**1.000E+02**	**1.002E+02**	1.232E+02	**1.000E+02**	**1.000E+02**
	SD	4.295E-13	2.422E-10	**1.435E-13**	7.414E+00	**1.435E-13**	9.65E-01

Table 3: Fitness errors of NAGGSA along with GSA variants over Testbed 2 (**TP** denotes Test Problem under consideration and **SD** stands for Standard Deviation)

TP	**metrics**	**GSA**	**GAGSA**	**PSOGSA**	**FVGGSA**	**PTGSA**	**NAGGSA**
$g_1(F1_{cec14})$	Mean	1.13E+08	1.78E+09	2.16E+08	4.03E+07	1.89E+08	**2.95E+07**
	Best	8.94E+07	1.413E+09	5.11E+07	2.57E+07	3.63E+07	**6.14E+06**
	SD	2.12E+07	2.41E+08	1.59E+08	**1.49E+07**	1.24E+08	1.88E+07
$g_2(F2_{cec14})$	Mean	9.81E+08	8.07E+10	1.41E+10	4.19E+08	1.25E+10	**7.05E+05**
	Best	6.65E+08	6.48E+10	1.45E+09	6.33E+03	5.01E+09	**1.99E+03**
	SD	3.07E+08	1.03E+10	1.76E+10	3.33E+08	5.79E+09	**3.48E+06**
$g_3(F3_{cec14})$	Mean	7.57E+04	8.51E+04	1.03E+05	6.84E+04	**7.79E+03**	9.73E+03
	Best	7.17E+04	8.39E+04	3.96E+04	5.34E+04	3.97E+03	**1.36E+03**
	SD	3.57E+03	**9.37E+02**	7.00E+04	6.41E+03	2.26E+03	4.81E+03
$g_4(F4_{cec14})$	Mean	2.89E+02	1.60E+04	9.44E+02	4.16E+02	**2.88E+02**	3.25E+02
	Best	2.57E+02	1.43E+04	2.46E+02	2.84E+02	2.05E+02	**1.54E+02**
	SD	**2.81E+01**	1.01E+03	8.52E+02	1.17E+02	3.18E+01	1.16E+02
$g_5(F5_{cec14})$	Mean	**2.00E+01**	2.11E+01	2.01E+01	**2.00E+01**	**2.00E+01**	**2.00E+01**
	Best	**2.00E+01**	2.10E+01	**2.00E+01**	**2.00E+01**	**2.00E+01**	**2.00E+01**
	SD	**1.45E-04**	4.88E-02	1.42E-01	3.28E-04	5.50E-04	5.03E-04
$g_6(F6_{cec14})$	Mean	2.75E+01	4.52E+01	**2.28E+01**	3.38E+01	2.38E+01	3.56E+01
	Best	2.36E+01	4.39E+01	1.96E+01	2.95E+01	**1.92E+01**	2.96E+01
	SD	2.86E+00	**1.06E+00**	2.03E+00	2.12E+00	2.55E+00	3.40E+00
$g_7(F7_{cec14})$	Mean	1.00E+01	7.60E+02	9.09E+01	1.55E+01	**1.51E-03**	1.93E-01
	Best	2.46E+00	6.39E+02	3.54E+01	2.07E+00	**1.90E-07**	1.35E-05
	SD	7.46E+00	8.10E+01	5.43E+01	8.72E+00	**4.05E-03**	3.71E-01
$g_8(F8_{cec14})$	Mean	1.44E+02	3.70E+02	**1.14E+02**	1.46E+02	1.44E+02	1.16E+02
	Best	1.37E+02	3.62E+02	1.00E+02	1.20E+02	1.16E+02	**9.25E+01**
	SD	6.87E+00	**5.26E+00**	1.40E+01	1.13E+01	1.08E+01	1.37E+01
$g_9(F9_{cec14})$	Mean	1.64E+02	3.46E+02	2.55E+02	1.77E+02	1.67E+02	**1.30E+02**
	Best	1.52E+02	3.30E+02	2.18E+02	1.50E+02	1.32E+02	**9.25E+01**
	SD	**1.25E+01**	1.45E+01	2.93E+01	1.64E+01	1.78E+01	1.95E+01
$g_{10}(F10_{cec14})$	Mean	3.73E+03	8.29E+03	4.33E+03	3.84E+03	**3.65E+03**	4.08E+03
	Best	3.22E+03	7.96E+03	3.33E+03	3.22E+03	2.91E+03	**2.65E+03**
	SD	3.57E+02	**2.50E+02**	5.97E+02	3.88E+02	4.32E+02	5.02E+02
$g_{11}(F11_{cec14})$	Mean	4.68E+03	8.73E+03	4.56E+03	4.54E+03	4.47E+03	**4.40E+03**
	Best	3.94E+03	8.24E+03	3.78E+03	3.61E+03	3.41E+03	**3.03E+03**
	SD	5.56E+02	**3.37E+02**	6.61E+02	5.10E+02	4.24E+02	6.19E+02

Table 3 Continued:

TP	metrics	GSA	GAGSA	PSOGSA	FVGGSA	PTGSA	NAGGSA
	Mean	1.47E+00	3.43E+00	1.43E-01	1.10E-02	**5.51E-03**	2.76E-01
$g_{12}(F12_{cec14})$	Best	1.32E+00	2.71E+00	7.45E-02	8.37E-04	**6.31E-04**	1.06E-01
	SD	1.66E-01	5.51E-01	6.37E-02	8.19E-03	**5.31E-03**	1.22E-01
	Mean	3.66E-01	9.16E+00	2.37E+00	4.70E-01	**3.28E-01**	4.42E-01
$g_{13}(F13_{cec14})$	Best	3.03E-01	8.76E+00	6.42E-01	2.76E-01	**2.25E-01**	3.02E-01
	SD	**4.08E-02**	3.60E-01	1.38E+00	3.49E-01	5.61E-02	8.54E-02
	Mean	1.58E+00	3.25E+02	6.34E+01	6.76E+00	2.49E-01	**2.42E-01**
$g_{14}(F14_{cec14})$	Best	2.18E-01	2.97E+02	3.19E+00	1.94E-01	1.70E-01	**1.60E-01**
	SD	3.00E+00	1.66E+01	6.59E+01	1.03E+01	**3.66E-02**	3.86E-02
	Mean	6.05E+01	4.45E+05	1.14E+05	3.17E+01	**2.25E+01**	1.07E+02
$g_{15}(F15_{cec14})$	Best	3.30E+01	3.94E+05	5.27E+01	2.13E+01	**9.81E+00**	7.32E+01
	SD	2.05E+01	3.52E+04	2.07E+05	**7.29E+00**	7.80E+00	2.17E+01
	Mean	1.35E+01	1.39E+01	1.31E+01	1.36E+01	1.36E+01	**1.29E+01**
$g_{16}(F16_{cec14})$	Best	1.29E+01	1.37E+01	1.25E+01	1.29E+01	1.31E+01	**1.23E+01**
	SD	3.87E-01	**1.90E-01**	4.91E-01	2.44E-01	2.20E-01	3.40E-01
	Mean	4.96E+06	1.77E+08	6.40E+06	5.83E+05	**2.75E+05**	3.65E+05
$g_{17}(F17_{cec14})$	Best	3.79E+06	7.08E+07	9.77E+04	2.78E+05	7.56E+04	**5.09E+04**
	SD	1.19E+06	8.51E+07	1.00E+07	**2.14E+05**	2.17E+05	3.13E+05
	Mean	6.21E+02	6.50E+09	8.80E+03	**4.81E+02**	1.04E+07	1.41E+03
$g_{18}(F18_{cec14})$	Best	2.59E+02	4.91E+09	4.92E+02	1.75E+02	1.48E+02	**9.11E+01**
	SD	5.12E+02	1.49E+09	1.00E+04	**3.01E+02**	5.28E+07	1.84E+03
	Mean	**6.78E+01**	6.19E+02	1.37E+02	1.60E+02	1.06E+02	8.86E+01
$g_{19}(F19_{cec14})$	Best	3.01E+01	5.26E+02	9.69E+01	4.56E+01	**2.35E+01**	3.70E+01
	SD	3.17E+01	5.41E+01	5.39E+01	3.95E+01	**3.15E+01**	3.57E+01
	Mean	1.58+05	7.76E+06	4.36E+04	5.18E+04	1.85E+04	**1.62E+04**
$g_{20}(F20_{cec14})$	Best	1.18E+05	9.36E+05	8.08E+03	3.93E+04	1.46E+04	**8.46E+03**
	SD	3.01E+04	1.34E+07	5.03E+04	8.29E+03	2.27E+03	**5.59E+03**
	Mean	1.63E+06	1.36E+08	1.96E+06	1.73E+05	1.41E+05	**1.14E+05**
$g_{21}(F21_{cec14})$	Best	4.88E+05	5.35E+07	5.68E+05	8.15E+04	6.84E+04	**1.03E+04**
	SD	7.64E+05	8.34E+07	2.04E+06	**4.82E+04**	4.23E+04	1.14E+05
	Mean	1.06E+03	5.63E+03	1.07E+03	1.21E+03	1.03E+03	**8.96E+02**
$g_{22}(F22_{cec14})$	Best	4.42E+02	2.76E+03	6.78E+02	4.16E+02	5.84E+02	**4.12E+02**
	SD	5.24E+02	4.27E+03	3.13E+02	3.26E+02	2.85E+02	**2.72E+02**

Table 4: p-values for comparison of NAGGSA with other considered GSA variants over Testbed 2

Post-hoc procedure	**GSA**	**GAGSA**	**PSOGSA**	**FVGGSA**	**PTGSA**
Holm	0.01004	$< 2E - 16$	$3.4E - 14$	$3.0E - 06$	0.83059
Hochberg	0.01004	$< 2E - 16$	$3.4E - 14$	$3.0E - 06$	0.83059
Hommel	0.01004	$< 2E - 16$	$3.4E - 14$	$3.0E - 06$	0.83059
Benjamin-Hochberg(BH)	0.00386	$< 2E - 16$	$8.5E - 15$	$7.4E - 07$	0.83059
Fdr	0.00386	$< 2E - 16$	$8.5E - 15$	$7.4E - 07$	0.83059