Enhanced Gravitational Search Algorithm Based on Improved Convergence Strategy

Norlina Mohd Sabri[1]*, Ummu Fatihah Mohd Bahrin[2] , Mazidah Puteh[3]
College of Computing-Informatics and Media,
Universiti Teknologi MARA Cawangan Terengganu, Kampus Kuala Terengganu, Malaysia

***Abstract*—Gravitational search algorithm (GSA) is one of the metaheuristic algorithms that has been popularly implemented in solving various optimization problems. The algorithm could perform better in highly nonlinear and complex optimization problems. However, GSA has also been reported to have a weak local search ability and slow searching speed to achieve its convergence. This research proposes two new parameters in order to improve GSA's convergence strategy by improving its exploration and exploitation capabilities. The parameters are the mass ratio and distance ratio parameters. The mass ratio parameter is related to the exploration strategy, while the distance ratio parameter is related to the exploitation strategy of the enhanced GSA (eGSA). These two parameters are expected to create a good balance between the exploration and the exploitation strategies in eGSA. There are seven benchmark functions that have been tested on eGSA. The results have shown that eGSA has been able to produce good performance in the minimization of fitness values and execution times, compared with two other GSA variants. The testing results have shown that the enhancements made to GSA have successfully improved the algorithm's convergence strategy. The improved convergence has also been able to improve the algorithm's solution quality and the processing time. It is expected that eGSA could be applied in many fields and solve various optimization problems efficiently.**

Keywords—Enhanced gravitational search algorithm; variant; improved convergence; exploration; exploitation

I. Introduction

Gravitational Search Algorithm (GSA) is a physics based metaheuristic algorithm which has been adapted to solve various optimization problems. The algorithm has been one of the popular optimization algorithms that has been adopted by the researchers [1]. GSA has been reported to have the capability to solve highly nonlinear and complex engineering optimization problems effectively [2-4]. Based on literatures, GSA has demonstrated better performance in solving optimization problems such as for constrained, unconstrained, continuous, discrete and multi objective optimizations [5, 6]. The algorithm has produced better performance than the other well-known algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) in solving various optimization problems [5, 7, 8].

Among of the advantages of GSA are such as simple concept, less control parameters and able to balance the exploration and exploitation in the optimization [9-12]. The exploration capability is related to the ability of the algorithm to expand the search space for good solutions. Algorithms must use exploration strategies at the beginning to avoid trapping in the local optimum. Exploitation in the metaheuristics is related to the convergence strategy or the ability to find the near optimal solution among good solutions. The strategies to achieve convergence are different for each of the algorithms. An optimal result is determined by the good balance between the exploration and exploitation capabilities.

As for GSA, the algorithm's exploitation characteristic has been reported to produce longer execution time to reach the optimal solution [13, 14]. The presence of agents with heavier masses at the end of run has contributed to longer computational time needed for the algorithm to reach the optimal solution. It has been reported that GSA suffers long computational time compared to other well-known algorithms. Based on literatures, the original GSA has been reported to have a poor local search ability and slow searching speed in the last iteration. The particles tend to get stuck in the local optima in the last stage of iterations. Due to the problem, the swarm cannot converge to the optimal point even though the particles could cluster to a small domain [15-17]. The algorithm requires more time to reach the optimal solution due to the presence of heavier masses at the end of run [18, 19].

Based on the reported problems, GSA has been suffering from the weak exploitation in certain problem domains. Due to the weakness, many enhancements and modifications have been done to GSA in order to improve its performance. Various GSA variants have been designed in order to improve its convergence rate, computational time and the solution quality. Some of the variants have focused on the modifications of the behavior of GSA by introducing new parameters or functions to the algorithm's structure. Among of the previously introduced parameters are L´evy flight operator and disruption operator [20,21]. The L´evy flight operator is designed to avoid the premature convergence, while the disruption operator improves the exploration and exploitation abilities of GSA. The other concepts are the adaptation of clustering method, stochastic local neighborhood search, chaos theory and natural selection rules [22-25]. These concepts have been introduced to reduce the complexity and computation of GSA and to avoid from local optima. The concepts which are related to physics theory such as astrophysics, mass dispersed gravity and wave function have also been introduced to the standard GSA in order to improve the algorithm's performance [26, 27, 21]. These modifications could enhance the global searching capability of GSA and also help the agent to escape from local optima.

The previous enhancements have considered the balance between the exploration and exploitation capabilities of the algorithm to achieve efficient searches. The good balance of the exploration and exploitation capabilities could be achieved by assigning the specific parameters or operators to the algorithm to produce specific capabilities [21]. Based on literatures, there are many additional parameters that have been introduced such as the escape velocity operator to improve the velocity [28], differential factor parameter to balance the global and local searches [29], chaotic perturbation operator to improve the exploitation [24] and disruption operator to avoid weaker agents [21]. GSA tends to get stuck in the last iteration after performing well at the beginning. Due to the problem, there is a need to add new operators into GSA in order to increase its efficiency in solving nonlinear optimization problems [30]. Hence, this research is proposing the enhancement of GSA with the introduction of mass ratio parameter and distance ratio parameter to the algorithm. The mass ratio parameter is designed to improve the exploration strategy, while the distance ratio parameter is designed to improve the exploitation strategy. The objective of the research is to improve the convergence of GSA by improving both of the exploration and exploitation capabilities of the algorithm. In this research, the original structure of GSA is modified to further improve its performance. The new parameters have been designed based on the basis that the parameters should be able to select only better agents as the active agents for the calculation of forces, while improving the exploration and exploitation capabilities. It is expected that this proposed convergence strategy could improve GSA in obtaining better optimal solution and improve its computational time.

This paper is organized into several sections. The earlier sections present the introduction and overview of GSA. The later sections explains the enhancements of GSA, the benchmark testing, results of the analyses and discussion. Finally the paper is summarized through the conclusion of the research.

II. Literature Review

A. Gravitational Search Algorithm (GSA)

Gravitational Search Algorithm (GSA) was developed by Rashedi et al. in 2009, which was based on Newton's law of gravity and law of motion [31]. In Newton's law of gravity, the force between two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between the objects. The second law of motion states that when a force is applied to an object, the acceleration is depending on the force and the mass. GSA is represented by agents, which carry their own masses in the search space. The following steps show the basic procedures of GSA:

1) *Randomized initialization.*
2) *Fitness evaluation of agents.*
3) *Update G(t), best(t), worst(t) and Mi(t) for i = 1,2,...,N.*
4) *Calculation of the total force in different directions.*
5) *Calculation of acceleration and velocity.*
6) *Updating agents' position.*
7) *Repeat steps 2 to 6 until the stopping criterion is reached.*

Based on the GSA procedures, the first step is the random initialization of the population of agents. The fitness values of the agents are evaluated in the second step. In the third step, the gravitational constant G(t) is updated based on the time execution, while the agents' masses, best and worst of the population are evaluated. In the fourth step, based on the evaluations, the total force in different direction of agents is calculated. The total force value leads to the updates of the acceleration and velocity of an agent as in the fifth step. Equation (1) and (2) show the computation of the acceleration a and the velocity v respectively. The acceleration a of an agent at iteration t is calculated based on the total force, Fid(t) and mass, Mii(t) as shown by (1). In the sixth step, the position value x of an agent is calculated based on (3) after the velocity value v has been obtained. After the maximum iteration has been reached, the execution would stop and return the optimal solution. In GSA, the biggest mass corresponds to the optimal solution while its position is the solution to the problem.

$$a_i^d(t) = F_i^d(t) / M_{ii}(t) \tag{1}$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{2}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{3}$$

III. Methodology

A. Enhancements of GSA

GSA has widely been improved since its introduction in 2009. Over the years, various GSA variants have been designed in order to improve its performance. The significant contributions in GSA have been focusing on improving the convergence rate, reducing computational efforts and improving the solution quality. Based on the previous studies on the improvement of GSA, the enhancements could be categorized into two approaches. One of the approaches is the modifications of the behavior of GSA by introducing new parameters or functions to GSA structure, while another approach is the hybridization of GSA with other intelligent techniques.

In this research, the exploration and the exploitation strategy of GSA is studied to overcome its convergence issues and improve the execution time. There are two new parameters that have been designed in the enhancements, namely mass ratio and distance ratio parameters. The mass ratio parameter is related to the exploration strategy, while the distance ratio parameter is related to the exploitation strategy of the enhanced GSA (eGSA).

B. Mass Ratio Parameter

The first enhancement of eGSA is the introduction of the mass ratio parameter, which is aimed to reduce the number of active agents in the search space. Based on the original concept of GSA, only *Kbest* agents should attract other agents to improve the algorithm's performance [31]. *Kbest* agents are the set of agents with better fitness values and bigger masses. In GSA, only *Kbest* agents should apply forces to the others to control the exploration and exploitation capabilities of the algorithm. This research is proposing a mass ratio parameter to select the set of *Kbest* agents in the search space. Based on the parameter, only agents with bigger masses will become active in attracting the other masses in the search space. The formula for the mass ratio parameter is shown in (4).

$$\text{Mass Ratio} = M_i / M_{best} \tag{4}$$

where:

M_i = mass of an agent

M_{best} = biggest mass in the search space

Based on (4), the mass ratio of an agent is obtained by dividing its mass with the biggest mass in the search space. The set of *Kbest* agents is the agents with mass ratio values in the interval [0.1, 1.0]. These *Kbest* agents would apply forces with each other within the given mass ratio values. This mass ratio approach is introduced to control the search for good candidate solutions in the search space based on the ratio of the biggest mass. Fig. 1 shows the concept of mass ratio approach among the agents. This approach is applying exploration at the beginning when all of the *Kbest* agents apply forces to each other. However, since *Kbest* is a function of time, its initial value of K_0 will decrease linearly with lapse of time. By the end of the iteration, there will be only one *Kbest* agent that applies force to the others. By the end of the iteration, exploitation is obtained by the decrement of the *Kbest* agents in the forces attractions. This mechanism could improve the exploration strategy as only the good solutions would be considered, eliminating the worse ones. The global search efficiency of GSA could be improved by selecting only better masses for the accumulation of forces. This approach could attain the good coordination between exploration and exploitation capabilities as the exploration would fade out when the exploitation starts to fade in.

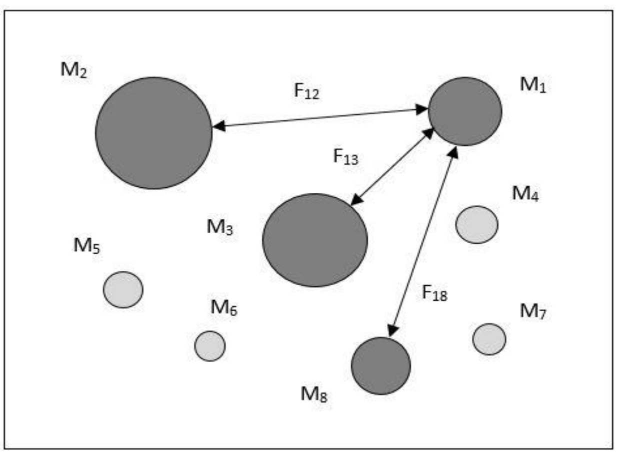

Fig. 1. Only agents with bigger masses apply forces to each other in eGSA.

C. Distance Ratio Parameter

The second enhancement of eGSA is the introduction of the distance ratio parameter. This enhancement is inspired by the reported GSA's local search weakness [32]. This second enhancement is based on the distance factor in the accumulation of forces between the agents. Based on the gravitational force formula as seen in the (5), the forces between two agents (F_{12}) are directly proportional to the gravitational constant, G and their masses (M_1 and M_2). The forces are indirectly proportional to their distance (R_2).

$$F_{12} = GM_1M_2 / R^2 \tag{5}$$

Based on (5), the force between the two agents (M_1 and M_2) would be higher when the distance between them is smaller. This formula has shown that an agent would have more attraction with other agents which are in shorter distances. Furthermore, the law of motion has stated that the force is directly proportional to the acceleration of the object. Equation (6) shows the acceleration, *a*, of an object is directly proportional to its force, *F* and indirectly proportional to its mass, *m*. Based on the equation, the smaller force would generate lower acceleration and vice versa.

$$a = F/M \tag{6}$$

In GSA, the acceleration is necessary to determine the velocity of an agent. Equation (2) has shown the formula for the velocity of the next iteration of an agent, $v_i(t+1)$ that is significantly influenced by the acceleration, $a_i(t)$. Based on (2), the low acceleration value would result in the lower velocity of an agent. Due to the lower velocity, the agent would continue their search in the bad area which would contribute to the decreased of the optimization result. Since the acceleration is depending on the force and the force is depending on the distance, the agents that are far away in the search space would generate lower velocities. The $rand_i$ is a uniform random variable which is in the interval of between 0 and 1. The purpose of the random number is to give a randomized characteristic to the search.

In this research, since the agents have already been selected through the mass ratio approach, the active agents in the search space are only the ones with the bigger masses. These agents are more efficient, have higher attractions and move slower than other lighter agents [31]. In order to help the agents to search the space more locally, the distance ratio parameter is introduced. Based on this parameter, only agents in shorter distance are selected for the accumulation of forces between agents. The accumulated force would be used to calculate the acceleration which would determine the velocity and position of an agent. The formula for the distance ratio is as shown in (7).

$$\text{Distance Ratio} = D_{i,j} / D_{i,max} \tag{7}$$

where:

$D_{i,j}$ = distance of an agent from other agent.

$D_{i,max}$ = the farthest distance with an agent.

Equation (7) has shown that the distance ratio is obtained by dividing the distance of an agent from another agent with the distance of the farthest agent in the search space. The distance of the farthest agent changes with each agent in the search space. In this research, the distance ratio is set in the interval of [0.1, 1.0]. The distance ratio approach is adapted into GSA in order to speed up the process of obtaining the forces between agents while improving the quality of solution. This approach would select agents in shorter distances depending on the ratio setting and this would help the algorithm to search the space more locally. This distance ratio parameter is introduced to help in the search for optimal solution or global optima more efficiently. Based on the literature, GSA requires more time to reach the optimal solution due to the presence of heavier masses at the end of run [33]. This approach would reduce the number of agents to be considered in the determination of the optimal solution at the end of run, hence could improve the algorithm's local search efficiency and is expected to improve the execution time.

D. Procedural Steps of eGSA

This section provides a more detailed description on the procedural steps of the enhanced GSA (eGSA) by showing the formula in each step of the algorithm. There are altogether 10 steps involved in order to obtain the final optimal result in each execution. The steps of the standard GSA has been briefly described in the earlier section. In eGSA, the concept of the algorithm is still based on the standard GSA, but with the additional enhancements to its structure. In the procedure, Step 4 to Step 7 is the new enhancement in eGSA. The following shows the procedural steps of eGSA:

Step 1: Agents initialization.

Step 2: Fitness evaluation, best and worst fitness computations.

Step 3: Gravitational constant (*G*) computation.

Step 4: Agent's Mass (M_i) and M_{best} computations.

Step 5: Selection of *Kbest* agents based on mass ratio parameter.

Step 6: Farthest distance, $D_{i,max}$ computation.

Step 7: Selection of agents based on distance ratio parameter.

Step 8: Accelerations of agents' computation based on total forces.

Step 9: Velocities and positions of agents' computation.

Step 10: Repeat steps 2 to 9.

The first step of eGSA is the agents' initialization in the search space. Equation (8) shows the first step of GSA which is to initialize the positions of the *N* number of agents.

$$X_i = (x_i^1,\ldots x_i^d, \ldots, x_i^k), \text{ for } i= 1,2,..,N. \quad (8)$$

Based on (8), x_i^d represents the positions of the i^{th} agent in the d^{th} dimension, while *k* is the space dimension.

The second step covers the computation of fitness evaluation for each agent, which led to the determination of the best and worst fitness among the agents. For example, the minimization function of GSA is selected. Equation (9) and (10) show the formula for the minimization problem.

$$best(t) = \min fit_j(t) \quad (9)$$

$$j \in \{1,\ldots,N\}$$

$$worst(t) = \max fit_j(t) \quad (10)$$

$$j \in \{1,\ldots,N\}$$

Based on (9) and (10), the $fit_j(t)$ represents the fitness value of the j^{th} agent at iteration *t*, *best(t)* and *worst(t)* represents the best and worst fitness at iteration *t*.

In the third step, the gravitational constant (*G*) is computed. Equation (11) shows the formula to calculate G, which is computed at iteration *t* [34].

$$G(t) = G0e(-\alpha t/T) \quad (11)$$

Based on (12), G_0 and α have to be initialized at the beginning and will be reduced with time to control the search accuracy. The *T* is the total number of iterations.

The fourth step is the computation of the agents' masses. In the theoretical physics, there are actually three kinds of masses that have been identified. The masses are the active gravitational mass, passive gravitational mass and the inertial mass. In GSA, the active, passive and inertia masses of an agent are considered to be equal based on the theory of the general relativity [32]. Based on (12), M_{ai} and M_{pi} are the active and passive gravitational masses respectively, while M_{ii} is the inertia mass of the i^{th} agent. Equation (12) shows that the three masses are actually equal. Equation (13) shows that the mass for each agent is calculated based on the worst and best fitness at the iteration *t*. Each of the mass *i* is then updated based on the other masses *j* as shown in the equation (14).

$$M_{ai} = M_{pi} = M_{ii} = M_i, \quad i = l, 2, \ldots ,N. \quad (12)$$

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

$$m_i(t) = fit_i(t) – worst(t) / best(t) – worst(t) \quad (13)$$

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

$$M_i(t) = m_i(t) / \Sigma_{j=1}\, m_j(t) \quad (14)$$

In this step, the calculation of masses for each of the agents has led to the determination of the best mass, M_{best}. In order to apply the mass ratio parameter, the determination of M_{best} has to be done in this step.

In the fifth step, the *Kbest* agents are selected based on the mass ratio parameter as shown in (4). Based on the mass ratio approach, only *Kbest* agents would become active and apply forces with each other in the search space. The sixth step is the calculation of the farthest distance, $D_{i,max}$ among the *Kbest* agents. This step is necessary in order to apply the distance ratio parameter in the next step.

The seventh step applies the other new distance approach parameter. In step 7, the active agents would be selected again based on the distance ratio parameter as shown in (7). Based on the distance ratio approach, only agents with shorter distances are selected for the calculation of forces between the agents.

The eighth step covers the calculation for the acceleration of agents. Before the calculation of the acceleration, the value for $F_{ij}^d(t)$ has to be computed based on (15). Based on (15), $F_{ij}^d(t)$ is the force acting on agent *i* from agent *j* at d^{th} dimension and t^{th} iteration. $R_{ij}(t)$ is the Euclidian distance between two agents *i* and *j* at iteration *t*. *G(t)* is the computed gravitational constant at the same iteration while ε is a small constant.

$$F_{ij}^d(t) = G(t).(M_{pi}(t) \times M_{aj}(t) / R_{ij}(t) + \varepsilon).(x_j^d(t) - x_i^d(t)) \quad (15)$$

After the calculation of $F_{ij}^d(t)$, only then the total force that acts on the i^{th} agent, $F_i^d(t)$ could be calculated based on (16). The total forces are calculated based on all of the agents that have been selected after the implementation of the distance ratio parameter.

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t) \quad (16)$$

The acceleration of the i^{th} agents at iteration t could be computed as already shown in (1). The ninth step covers the calculations for the velocity, v_i and position, x_i of each agent. The velocity and the position of the agents at the next iteration $(t+1)$ are computed based on (2) and (3) respectively. The $rand_i$ is the random variable in the interval [0,1] which would give the randomized characteristic to the search.

The final step is to repeat the step 2 to step 9 until the iterations reach the maximum limit. The best fitness value at the final iteration is computed as the global fitness while the position of the corresponding agent is computed as the global solution of this problem.

Based on these enhancements, the new flowchart for eGSA is shown in Fig. 2. The highlighted parts in the Fig. 2 show the enhancement that has been implemented in eGSA.

E. Benchmark Functions Testing

The enhanced GSA (eGSA) has been tested with seven benchmark test functions in order to validate its capabilities. The selected functions are the commonly used benchmark functions that have been applied to test the performance of an optimization algorithm [31, 35, 36]. The function names, their mathematical representations, characteristics and the search spaces are given in Table I.

TABLE I. BENCHMARK FUNCTIONS APPLIED IN THE EXPERIMENTS

Function Name	Mathematical Representation	Charac-teristic	Search Space
Sphere	$F_1(X) = \sum_{i=1}^{n} x_i^2$	Unimodal	$[-100,100]^n$
Schwefel 2.21	$F_2(X) = \max\{\lvert x_i \rvert, 1 \le i \le n\}$	Unimodal	$[-100,100]^n$
Step	$F_3(X) = \sum_{i=1}^{n}([x_i + 0.5])^2$	Unimodal	$[-100,100]^n$
Quartic Noise	$F_4(X) = \sum_{i=1}^{n} i\, x_i^4 + random[0,1]$	Unimodal	$[-.28,1.28]^n$
Rosenbrock	$F_5(X) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	Multimodal	$[-30,30]^n$
Schwefel 2.26	$F_6(X) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	Multimodal	$[-500,500]^n$
Rastrigin	$F_7(X) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	Multimodal	$[-.12,5.12]^n$

* n = dimension

Based on Table I, four of the functions are the unimodal functions while another three are the multimodal functions. The unimodal functions are the functions with only one single local minima and are commonly applied to test the convergence rate of a search. As for the multimodal, the functions have many local minima and are commonly applied to test the ability of the algorithm to escape from the local optima. The final result in the multimodal function is important as it shows the ability of an algorithm to find the global optima. In this benchmark testing, eGSA has been compared with another two modified or variants of GSA algorithms. The other two variants of GSA algorithms have been selected based on their almost similar concepts for enhancement with eGSA. Both of the algorithms have also been based on the distance of agents for their points of change in the GSA's structure. The comparison algorithms are the Improved GSA (IGSA) and Hybrid Gravitational Search with Lévy Flight (HGSLF) [37, 38]. The IGSA has been based on the disruption phenomena in the outerspace, where a star of the system could disrupt other objects under the influence of its gravitational force. In the IGSA algorithm, an agent is disrupted if the ratio of the distance between its mass and the neighbouring mass to its distance from the best solution is smaller than a specified threshold. As for the Lévy flight operator, it is applied to one of the mass if the distance between the two masses have become very near and both of them are not good solutions in the search space.

The parameter settings for each of the algorithm have been provided in the Table II to Table IV respectively. Based on the tables, the standard GSA parameters are the gravitational initial value, alpha and epsilon. The value of G_o and α determine the convergence speed and help to balance the exploration and exploitation of GSA [39]. As for the epsilon, ε, it helps in the updating strategy of GSA. The other parameter settings are for the new introduced parameters, which are the mass ratio and distance ratio for eGSA, the constant operator θ and small value ρ for IGSA and the treshold constant ξ for HGSLF. These new parameters would determine the exploration or exploitation capabilities of the algorithms respectively.

TABLE II. PARAMETER SETTING FOR EGSA

Parameter	Value
Gravitational initial value, G_o	100
Alpha, α	20
Epsilon, ε	0.00001
Mass ratio	0.1
Distance ratio	0.9

TABLE III. PARAMETER SETTING FOR IGSA

Parameter	Value
Gravitational initial value, G_o	100
Alpha, α	20
Epsilon, ε	0.0001
θ (constant operator)	100
ρ (small value)	10^{-16}

TABLE IV. PARAMETER SETTING FOR HGSLF

Parameter	Value
Gravitational initial value, G_o	100
Alpha, α	20
Epsilon, ε	0.0001
ξ (treshold constant)	10^{-3}

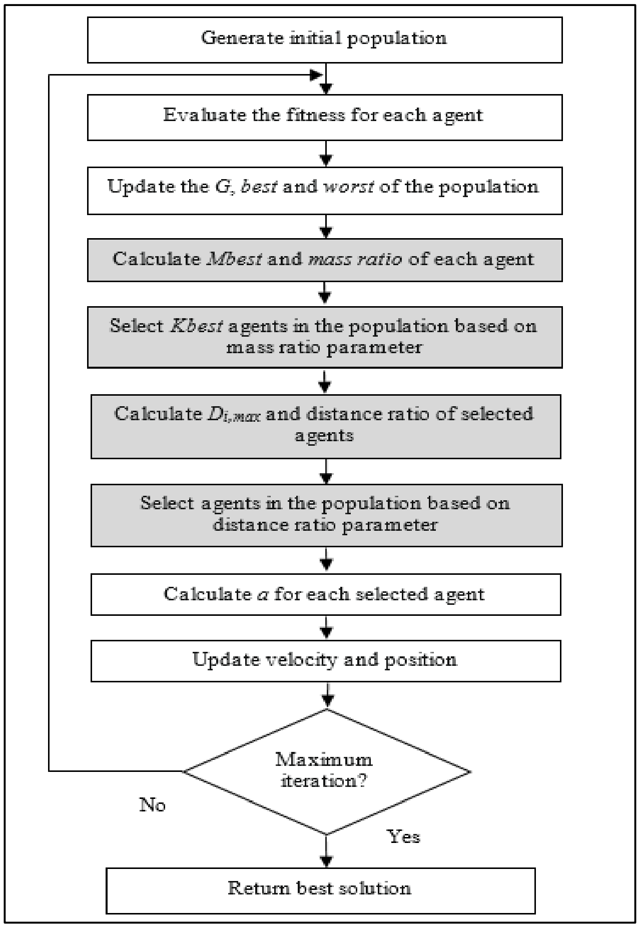

Fig. 2. Flowchart of the enhanced GSA (eGSA). Adapted from (Rashedi et al.,2009).

IV. Result and Discussion

The experimental results in this research have been divided into the fitness values and processing times evaluations. These performance measurements have been evaluated based on the statistical analyses.

A. Fitness Values

This section provides the fitness value analyses from the results of the benchmark testing between eGSA and the other two comparison algorithms, IGSA and HGSLF. In the benchmark functions testing, each of the dimension of the function is 30 (n=30), the population size is 50 (N=50) and the maximum iteration (t_{max}) has been set to 1000. Based on Table I, the minimum values (f_{opt}) for all of the 7 functions are 0, except for F_6 which has a minimum value of -418.9829 x n. The minimization results of the benchmark functions testing for each of the algorithm are shown in the following Table V. The best, worst and the mean fitness of the solutions, which have been averaged over 30 runs have been recorded in the table.

Based on Table V, the performance of eGSA is acceptable in all of the seven functions. Based on the overall results, eGSA is able to minimize the unimodal and multimodal functions. The mean fitness values for F1 (0.4822) and F2 (0) have been able to reach 0, while for F3 (6.6875), F4 (4.2713) and F5 (1.7320), the mean fitness values are almost reaching 0 values. For F6 and F7, these multimodal functions have many local optima and are difficult to optimize. However, eGSA has been able to minimize the functions and the results are satisfying. Based on Table V, the overall results show that the performance of eGSA is better than IGSA and HGSLF in almost all of the functions.

TABLE V. Minimization Results of Benchmark Functions

Test Function		eGSA	IGSA	HGSLF
F1	Best	0.0191	20224.06	16124.47
	Worst	1.4021	28298.42	24884.73
	Mean	0.4822	24072.19	20618.17
F2	Best	0	0	0
	Worst	0	0	0
	Mean	0	0	0
F3	Best	6.0192	78.2867	17771.13
	Worst	7.5000	306.7619	22977.62
	Mean	6.6875	147.2939	20447.29
F4	Best	1.3216	72.7237	69.0424
	Worst	7.7193	137.6973	85.0527
	Mean	4.2713	104.4885	78.1088
F5	Best	0.2022	305.2722	472.9982
	Worst	3.4489	342.4960	613.2155
	Mean	1.7320	324.3189	538.8112
F6	Best	-3360.7363	-2598.1066	-3937.7373
	Worst	-2852.5944	-2187.7641	-3586.5394
	Mean	-3150.2839	-2396.7062	-3780.7486
F7	Best	270.7417	378.7011	379.4766
	Worst	293.9898	445.3634	391.2795
	Mean	278.5284	414.4412	385.0295

The performance of eGSA, IGSA and HGSLF for the minimizations of the unimodal functions F1 to F4 have been illustrated in Fig. 3 to Fig. 6. The figures show that eGSA is able to minimize and is able to converge with better mean fitness values compared to the other algorithms. As for IGSA and HGSLF, the algorithms still have been able to minimize and converge with larger values in most of the functions. However, IGSA has not been able to further minimize the results in F1 and F4. This is due to the decrement of the values in the minimization that have been very small, which is in decimal point values. In the early iteration of IGSA, most of the agents have been disrupted and their position values have been changed to become much smaller due to the multiplication with the D value.

In this research, the algorithms have been coded using Java for experimental purposes. Java has some limitations such as limited floating point representation and the random numbers are generated based on the pseudorandom numbers. The initial seeding of the population is important as it would affect the final results. However, the HGSLF is still able to minimize most of the benchmark functions. In this research, it is the IGSA that has difficulties in the function minimizations, most probably due to its more complex additional structure and also due to Java limitation in the floating point representation.

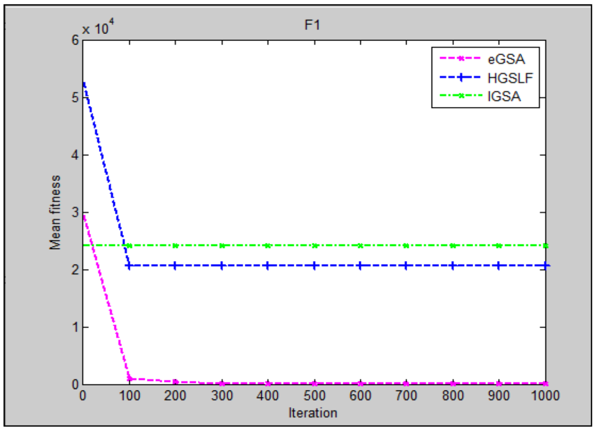

Fig. 3. Performance of the algorithms in the minimization of F1.

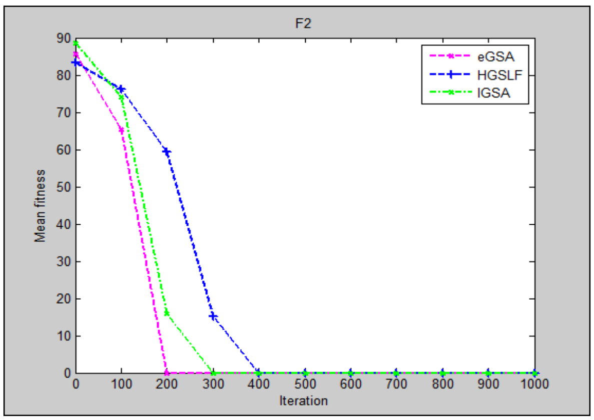

Fig. 4. Performance of the algorithms in the minimization of F2.

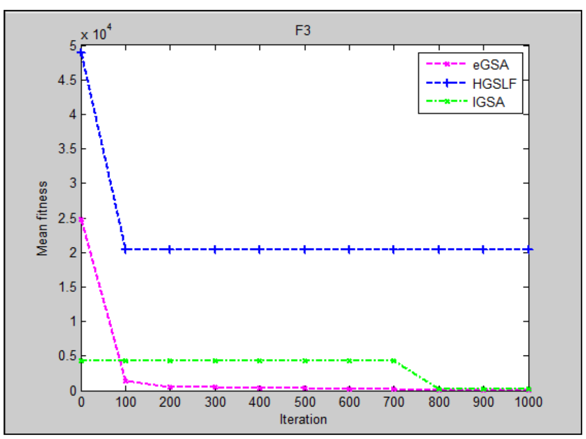

Fig. 5. Performance of the algorithms in the minimization of F3.

As for the multimodal functions, the performance results have been illustrated from Fig. 7 to Fig. 9. Fig. 7 shows that all of the algorithms have been able to minimize and obtain the f_{min} values of 0 for function F5. Fig. 8 and Fig.9 show that for functions F6 and F7, eGSA and HGSLF have been able to minimize and have obtained acceptable mean fitness values. However, IGSA is unable to further minimize as it tends to trap in the local optima as shown in the F6 and F7 results. In this experimental study, it is difficult for IGSA to search for the global optimum in the minimization of the F6 and F7 functions. This is also due to the very small decrement values in the minimization of the functions.

In this experimental study, the results of IGSA and HGSLF were not as good as that had been previously reported. The reported previous results have been tested using Matlab which has limitless floating point numbers. However, in this research, Java has been selected and used for experimental purposes compared to the standard Matlab tool. Java is also a popular, powerful and robust programming language that has been implemented in various applications. This research has shown that Java could also be used for the minimization of test functions for optimization problems.

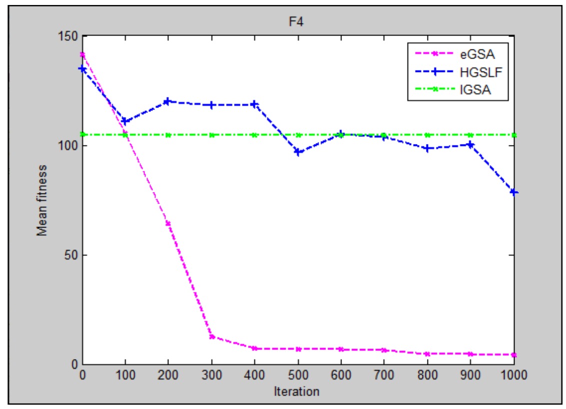

Fig. 6. Performance of the algorithms in the minimization of F4.

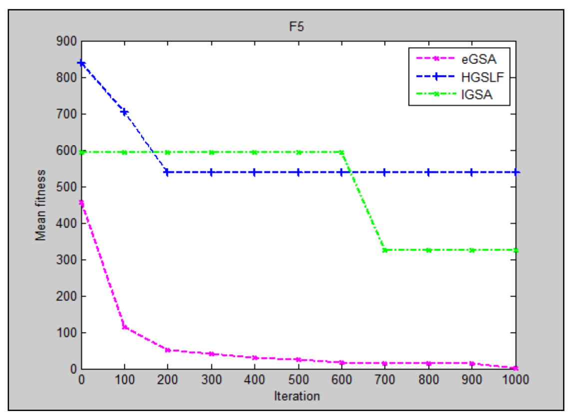

Fig. 7. Performance of the algorithms in the minimization of F5.

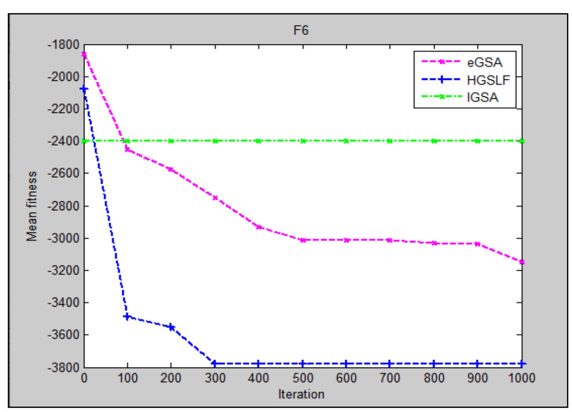

Fig. 8. Performance of the algorithms in the minimization of F6

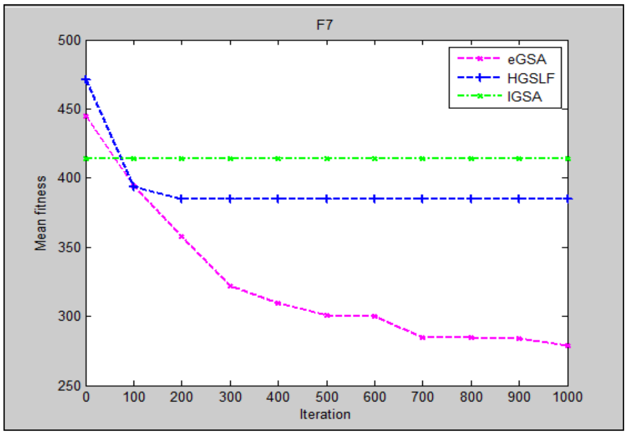

Fig. 9. Performance of the algorithms in the minimization of F7

B. *Processing Times*

This section provides the processing times taken by eGSA, IGSA and HGSLF to minimize all of the seven benchmarks functions in this experimental study. Table VI shows the processing times for the seven benchmark test functions.

TABLE VI. PROCESSING TIMES FOR BENCHMARK TEST FUNCTIONS (MS)

Test Function		eGSA	IGSA	HGSLF
F1	Min	671	3686	2215
	Max	814	3809	5042
	Mean	**732**	3745	3489.75
F2	Min	239	2883	2711
	Max	433	4435	4582
	Mean	**314.40**	3780	3543.25
F3	Min	615	3836	2702
	Max	650	3926	4618
	Mean	**633.67**	3871.25	3542.25
F4	Min	609	2448	4088
	Max	689	2576	6197
	Mean	**646.25**	2489	5280.75
F5	Min	800	3726	2873
	Max	994	4031	5353
	Mean	**898.33**	3866	3891.25
F6	Min	4707	4566	3932
	Max	5749	6095	4763
	Mean	5304.20	5242.75	**4328.50**
F7	Min	3822	3975	3885
	Max	4283	6021	5537
	Mean	**3979**	4845	4508.33

Based on Table VI, the mean processing times of eGSA for most of the functions, except F6, have been the lowest compared to the other 2 algorithms. These faster processing times have been due to the eGSA concept which is to reduce the number of active agents in the search space. In eGSA, the active agents would be selected initially based on the mass ratio and then would further be selected based on the distance ratio. Table VII shows the average number of active agents in eGSA over the 30 runs after the mass ratio and distance ratio parameters have been applied in the search space. Based on the table, it could be seen that the number of active agents have been reduced from the initial number of 50 after the mass ratio parameter have been applied. These active agents are the agents with bigger masses and represent good solutions in the search space. This mass ratio has been applied to improve the exploration of good solutions in the search space. In order to further improve the solution quality and the processing times, the active agents are further selected for the accumulation of forces between agents. Thus, the number of active agents would further be decreased after the implementation of the distance ratio parameter. This distance ratio parameter has been applied to improve the exploitation capability of eGSA. Based on the distance ratio, only the neighboring agents within the specified ratio would be selected for the accumulation of forces.

TABLE VII. AVERAGE NUMBER OF ACTIVE AGENTS BASED ON EGSA CONCEPT

	F1	F2	F3	F4	F5	F6	F7
Mass ratio	11	21	10	23	18	23	25
Distance ratio	8	16	7	20	14	19	21

As for IGSA and HGSLF, the number of active agents in the search space would not be reduced. Based on their respective concepts, only the positions of the related agents would be changed in the search space. Thus, the processing times of both of the algorithms would not be reduced in this functions minimizations. In both of the algorithms, the position changes have been designed in order to further improve especially in the exploration capabilities of the algorithms.

C. *Discussion*

In the benchmark function testing, eGSA has been able to minimize and produce acceptable results. Based on the testing, the proposed enhancements have been able to improve the algorithm's convergence strategy. In the minimization results, the mean fitnesses of eGSA in almost all of the test functions are better than IGSA and HGSLF. The execution time of eGSA are also lesser than the other two variants in all of the test functions testing. This shows that the introduction of the two new parameters has been able to improve on the exploration and exploitation capabilities of the algorithm [30]. The mass ratio parameter would improve the exploration, in which the algorithm would select only the good solutions based on the mass ratio in the search space and eliminate the weaker solutions. After the exploration, the distance ratio parameter would take over to improve the exploitation capability of the algorithm. Based on the distance ratio, only the nearest agents would be selected in finding the optimal solution. Due to this approach, the search space would become smaller in scale and the algorithm would be inclined to search more locally [31]. These two parameters are expected to create a good balance between the exploration and the exploitation strategies in the enhanced algorithm. It is expected that the enhanced GSA (eGSA) could achieve both the efficient global and local searches in order to improve its convergence to optimal solution. This would help in obtaining better solution quality and reduce the execution time in solving real world optimization problems.

V. Conclusion

This paper has discussed on the improvement of GSA convergence strategy, which is based on the two enhancements. The first enhancement is to assign *Kbest* agents based on the mass ratio parameter. This approach could filter and reduce the number of active agents in the search space. The second enhancement is the implementation of the distance ratio parameter to select only the nearest agents for the accumulation of forces among the *Kbest* agents. This second approach would reselect the agents based on the distance in order to further improve the execution time and also improve the solution quality.

The contribution of the research is the introduction of a new variant of GSA, namely enhanced GSA (eGSA) which could improve the algorithm's convergence strategy. Improved convergence strategy is expected to improve the performance of GSA in terms of its solution quality and computational time. In this research, eGSA has been designed mainly to reduce the number of active agents for the accumulation of the gravitational forces. The mass ratio and distance ratio operators have been designed to select only the bigger masses which represent the best solutions in the search space. It is expected that eGSA could improve the exploration and exploitation capabilities compared to the standard GSA and other variants. Significantly, eGSA has been able to perform better than two other GSA variants in the benchmark testing. The conclusion that could be derived based on the testing results is that the enhancement made to GSA has been successfully improve the algorithm's convergence, thus improving its solution quality and the processing time. The enhancements in the exploration and exploitation strategies of eGSA has enabled the algorithm to produce better results. The benchmark function testing results have shown that eGSA could produce good performance in solving minimization problems.

In future, the research on the enhancements or modifications of GSA would continue to expand as GSA has increasingly gained attentions due to its acceptable performance in solving various optimization problems. Besides, currently there are various real world optimization problems that need to be explored and solved using metaheuristics approaches.

Acknowledgment

Special gratitude goes to Universiti Teknologi MARA Cawangan Terengganu for the continuous support given to the advancement of research and publication in the university. This research is funded under the UiTM Geran Penyelidikan Myra Lepasan PHD (600-RMC/GPM LPHD 5/3 (068/2021).

References

[1] R. Guha, M. Ghosh, A. Chakrabarti, R. Sarkar, and S. Mirjalili, "Introducing clustering based population in binary gravitational search algorithm for feature selection," Applied Soft Computing, vol. 93, p. 106341, 2020.

[2] S. K. Joshi, "Levy flight incorporated hybrid learning model for gravitational search algorithm," Knowledge-Based Systems, vol. 265, p. 110374, 2023.

[3] Y. Liu, X. Zhang, and H. Chao, "An improved gravitational search algorithm combining with centripetal force," Partial Differential Equations in Applied Mathematics, vol. 5, p. 100378, 2022.

[4] Z.-k. Feng, S. Liu, W.-j. Niu, S.-s. Li, H.-j. Wu, and J.-y. Wang, "Ecological operation of cascade hydropower reservoirs by elite-guide gravitational search algorithm with Lévy flight local search and mutation," Journal of Hydrology, vol. 581, p. 124425, 2020.

[5] Z. Lei, S. Gao, S. Gupta, J. Cheng, and G. Yang, "An aggregative learning gravitational search algorithm with self-adaptive gravitational constants," Expert Systems with Applications, vol. 152, p. 113396, 2020.

[6] G. Tian, A. M. Fathollahi-Fard, Y. Ren, Z. Li, and X. Jiang, "Multi-objective scheduling of priority-based rescue vehicles to extinguish forest fires using a multi-objective discrete gravitational search algorithm," Information Sciences, vol. 608, pp. 578-596, 2022.

[7] R. Shanker and M. Bhattacharya, "An automated computer-aided diagnosis system for classification of MR images using texture features and gbest-guided gravitational search algorithm," Biocybernetics and Biomedical Engineering, vol. 40, no. 2, pp. 815-835, 2020.

[8] A. Naserbegi and M. Aghaie, "Multi-objective optimization of hybrid nuclear power plant coupled with multiple effect distillation using gravitational search algorithm based on artificial neural network," Thermal Science and Engineering Progress, vol. 19, p. 100645, 2020.

[9] Q. S. Banyhussan, A. N. Hanoon, A. Al-Dahawi, G. Yıldırım, and A. A. Abdulhameed, "Development of gravitational search algorithm model for predicting packing density of cementitious pastes," Journal of Building Engineering, vol. 27, p. 100946, 2020.

[10] F. Zhao, F. Xue, Y. Zhang, W. Ma, C. Zhang, and H. Song, "A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution," Expert Systems with Applications, vol. 113, pp. 515-530, 2018.

[11] H. Mittal and M. Saraswat, "An automatic nuclei segmentation method using intelligent gravitational search algorithm based superpixel clustering," Swarm and Evolutionary Computation, vol. 45, pp. 15-32, 2019.

[12] B. Yin, Z. Guo, Z. Liang, and X. Yue, "Improved gravitational search algorithm with crossover," Computers & Electrical Engineering, vol. 66, pp. 505-516, 2018.

[13] T. A. Khan and S. H. Ling, "A novel hybrid gravitational search particle swarm optimization algorithm," Engineering Applications of Artificial Intelligence, vol. 102, p. 104263, 2021.

[14] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, and Y. Deng, "Improving exploration and exploitation via a hyperbolic gravitational search algorithm," Knowledge-Based Systems, vol. 193, p. 105404, 2020.

[15] A. Guo, Y. Wang, L. Guo, R. Zhang, Y. Yu, and S. Gao, "An adaptive position-guided gravitational search algorithm for function optimization and image threshold segmentation," Engineering Applications of Artificial Intelligence, vol. 121, p. 106040, 2023.

[16] N. Aditya and S. S. Mahapatra, "Switching from exploration to exploitation in gravitational search algorithm based on diversity with Chaos," Information Sciences, vol. 635, pp. 298-327, 2023.

[17] D. Kumar and M. Rani, "Alternated superior chaotic variants of gravitational search algorithm for optimization problems," Chaos, Solitons & Fractals, vol. 159, p. 112152, 2022.

[18] W.-j. Niu, Z.-k. Feng, and S. Liu, "Multi-strategy gravitational search algorithm for constrained global optimization in coordinative operation of multiple hydropower reservoirs and solar photovoltaic power plants," Applied Soft Computing, vol. 107, p. 107315, 2021.

[19] J. Jiang, R. Jiang, X. Meng, and K. Li, "SCGSA: A sine chaotic gravitational search algorithm for continuous optimization problems," Expert Systems with Applications, vol. 144, p. 113118, 2020.

[20] A. F. Ali, "A Hybrid Gravitational Search with Levy Flight for Global Numerical Optimization," Inf. Sci. Lett., vol. 83, no. 2, pp. 71–83, 2015.

[21] S. Sarafrazi, H. Nezamabadi-pour, and S. Saryazdi, "Disruption: A new operator in gravitational search algorithm," Sci. Iran., vol. 18, no. 3, pp. 539–548, Jun. 2011, doi: 10.1016/j.scient.2011.04.003.

[22] M. Shams, E. Rashedi, and A. Hakimi, "Clustered-gravitational search algorithm and its application in parameter optimization of a low noise amplifier," Appl. Math. Comput., vol. 258, pp. 436–453, 2015, doi: 10.1016/j.amc.2015.02.020.

[23] T. Chakraborti, K. Das, and A. Chatterjee, “A novel local extrema based gravitational search algorithm and its application in face recognition using one training image per class,” Eng. Appl. Artif. Intell., vol. 34, pp. 13–22, 2014, doi: 10.1016/j.engappai.2014.05.002.

[24] S. Jiang, Y. Wang, and Z. Ji, “Convergence analysis and performance of an improved gravitational search algorithm,” Appl. Soft Comput., vol. 24, pp. 363–384, 2014, doi: 10.1016/j.asoc.2014.07.016.

[25] Y. Chen, H. Duan, and S. Member, “Multiple UCAVs Mission Assignment Based on Modified Gravitational Search *,” 2014.

[26] M. Davarynejad, J. Van Den Berg, and J. Rezaei, “Evaluating center-seeking and initialization bias : The case of particle swarm and gravitational search algorithms,” Inf. Sci. (Ny)., vol. 278, pp. 802–821, 2014, doi: 10.1016/j.ins.2014.03.094.

[27] M. Soleimanpour-moghadam and H. Nezamabadi-pour, “An improved quantum behaved gravitational search algorithm,” in 20th Iranian Conference on Electrical Engineering, (ICEE2012), 2012, no. 4, pp. 711–714.

[28] U. Güvenç and F. Katırcıoğlu, “Escape velocity: a new operator for gravitational search algorithm,” Neural Comput. Appl., pp. 1–16, 2017.

[29] S. Deepa and J. Rizwana, “Minimization of losses and FACTS installation cost using proposed differential gravitational search algorithm optimization technique,” J. Vib. Control, no. October 2014, 2015, doi: 10.1177/1077546315576612.

[30] H. Garg, “A hybrid GSA-GA algorithm for constrained optimization problems,” Inf. Sci. (Ny)., vol. 478, pp. 499–523, 2019, doi: 10.1016/j.ins.2018.11.041.

[31] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," Information sciences, vol. 179, no. 13, pp. 2232-2248, 2009.

[32] L. Ling-Ling, L. Guo-Qian, T. Ming-Lang, T. Kimhua, and L. Ming, "A maximum power point tracking method for PV system with improved gravitational search algorithm," Applied Soft Computing, vol. 65, pp. 333-348, 2018.

[33] S. Mallick, S. Ghoshal, P. Acharjee, and S. Thakur, "Optimal static state estimation using improved particle swarm optimization and gravitational search algorithm," International Journal of Electrical Power & Energy Systems, vol. 52, pp. 254-265, 2013.

[34] A. Chatterjee, G. Mahanti, and P. R. S. Mahapatra, "Generation of phase-only pencil-beam pair from concentric ring array antenna using gravitational search algorithm," in 2011 International Conference on Communications and Signal Processing, 2011: IEEE, pp. 384-388.

[35] S. He, L. Zhu, L. Wang, L. Yu, and C. Yao, "A modified gravitational search algorithm for function optimization," IEEE Access, vol. 7, pp. 5984-5993, 2019.

[36] M. Davarynejad, J. van den Berg, and J. Rezaei, "Evaluating center-seeking and initialization bias: The case of particle swarm and gravitational search algorithms," Information Sciences, vol. 278, pp. 802-821, 2014.

[37] S. Sarafrazi, H. Nezamabadi-pour, and S. Saryazdi, "Disruption: a new operator in gravitational search algorithm," Scientia Iranica, vol. 18, no. 3, pp. 539-548, 2011.

[38] A. F. Ali, "A hybrid gravitational search with levy flight for global numerical optimization," Information Sciences Letters Inf. Sci. Lett, vol. 4, pp. 71-83, 2015.

[39] G. Sun, P. Ma, J. Ren, A. Zhang, and X. Jia, “A stability constrained adaptive alpha for gravitational search algorithm,” Knowledge-Based Syst., vol. 139, pp. 200–213, 2018, doi: 10.1016/j.knosys.2017.10.018.