Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 345–359
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6248

Gravitational evaluation algorithm for global optimization problem

Hasanain Jalil Neamah[a,b,c,*], Ali M. Almobarqaa[d], Zainab Ali Abdulhusien[a]

[a] *Faculty of Management, University of Warsaw, Poland*

[b] *University of Information Technology and Communications, Baghdad, Iraq*

[c] *Acharya Nagarjuna University, Guntur, India*

[d] *Consultor of Development and Social Justice, Iraqi Parliament Council, Baghdad, Iraq*

(Communicated by Madjid Eshaghi Gordji)

Abstract

This work proposes a new metaheuristic technique that combines Differential evolution (DE) with gravity search in a consistent manner. Swarm intelligence benefits and the concept of tensile strength between two particles are combined to suggest superior meta-heuristic approaches for limitless optimization issues. The goal of this paper is to create a new algorithm that overcomes the shortcomings of the Gravitational search algorithm by leveraging the advantages of the Differential evolution algorithm in expanding search areas, overcoming early convergence problems, and improving the attractive algorithm's ability to converge towards the optimum. The GSA algorithm has been utilized in a search-oriented algorithm, whereas the Differential evolution algorithm is causing a high level of diversification in society, which leads to the establishment of search regions for the GSA algorithm. The effectiveness of the suggested approach was evaluated by solving a collection of 30 Real-Parameter Numerical Optimization problems that were presented at IEEE-CEC 2014. The findings are compared to 5 state-of-the-art unconstrained problem algorithms and 6 state-of-the-art unconstrained problem algorithms. The winner methods were also deduced from the results using the Wilcoxon signed test.

Keywords: gravitational search algorithm, global optimization, differential evolution, hybrid algorithms
2020 MSC: 38A68, 38W50, 83B05

1 Introduction

In the modern day, meta-heuristic algorithms have taken over as the primary method for solving optimization issues. As a whole, meta-heuristics may be broken down into three primary categories: physics-based, SI, and Evolutionary approaches (EAs).

In physics-based procedures, for example. Such optimization methods are often modeled after physical laws and principles. The algorithms Simulated Annealing, Harmony search, Big-Bang Big-Crunch (BBBC)[5, 29, 15, 7, 2, 10, 14, 34] and Curved The mechanism of these algorithms differs from that of EAs in that a random group of search

*Corresponding author

Email addresses: hasanien.1975@uoitc.edu.iq (Hasanain Jalil Neamah), alialmobarqaa@yahoo.com (Ali M. Almobarqaa), zainabalsaedi1983@gmail.com (Zainab Ali Abdulhusien)

Received: December 2021 *Accepted:* February 2022

agents interact with one another and move about in the search space according to physical laws rather than using a computer algorithm. This movement is accomplished by the use of several forces, such as gravity force, ray casting, electromagnetic force, inertia force, and weights, among others. For example, the BBBC algorithm was influenced by the hypotheses of the big bang and the big crunch. According to the concepts of the big bang theory, the search agents of BBBC are dispersed over a search space in random directions starting from a single point in the search space. According to the ideas of the big crunch theory, they seek at random and then congregate around a final location (which is the best spot they have found so far). GSA is another another algorithm that is based on physics. Newton's law of universal gravitation is the fundamental scientific theory from which GSA draws its inspiration. The GSA method searches for a solution by deploying a collection of agents with masses proportionate to the value of a fitness function in order to find it. When the masses are drawn to one other by the gravitational forces that exist between them, this is known as iteration. The attracting force increases in proportion to the mass of the object. The most massive mass draws the other masses in proportion to their distances from it, indicating that it is probably near to finding the global optimum.

In the case of SI algorithms. They are primarily designed to imitate natural swarming, herding, flocking, or schooling behavior in animals and other organisms. Even though the technique is very identical to that of a physics-based algorithm, the search agents move around the environment utilizing the simulated collective and social intelligence of animals. The PSO approach is the most often used SI technique. In their paper [16], Kennedy and Eberhart suggest a PSO algorithm that is based on the social behavior of flocking birds. The PSO algorithm makes use of numerous particles that pursue the location of the best particle as well as their own best positions earned so far in the game. In other words, a particle gets relocated based on the best solution it has found for itself as well as the best solution the swarm has found for itself. Dorigo et al. introduced the ACO algorithm in 2006 [4], which is another prominent SI technique. The social behavior of ants in an ant colony served as inspiration for the design of this program. It is in fact the social intelligence of ants in determining the shortest route between their colony and a food supply that has served as the primary inspiration for ACO. Candidate solutions are responsible for the evolution of a pheromone matrix during the duration of iteration. The ABC algorithm is another widely used algorithm that is based on the collective behavior of bees in their search for food sources. In ABS, there are three categories of bees: scout bees, observer bees, and hired bees, to name a few. The scout bees are in charge of investigating the search area, whilst the spectator and hired bees are in charge of capitalizing on the potential solutions discovered by the scout bees. The Bat-inspired Algorithm (BA), which is based on the echolocation activity of bats, was recently suggested [37], and it is a last alternative. In the wild, there are many different kinds of bats. They are all distinct in terms of size and weight, but when it comes to navigation and hunting, they all behave in a very similar manner. In order to do this, bats make use of their inherent sonar. The two most important traits of bats while hunting for prey have been included into the development of the BA algorithm. When chasing prey, bats have a tendency to reduce the volume of the ultrasonic sound they make while increasing the pace at which it is produced. The BA method has been theoretically developed to account for this tendency. More importantly, there have been other EA strategies presented to date, many of which are influenced by hunting and seeking behaviors, such as In 2001, the Honey Bees Optimization Algorithm (MBO) was modified to allow for marriage [1]. The Artificial Fish-Swarm Algorithm (AFSA) was developed in 2003 [20], while the Termite Algorithm was developed in 2005 [31]. In 2007, the Wasp Swarm Algorithm was developed [28]. In 2007, there was a monkey search [26]. In 2008, the Bee Collecting Pollen Algorithm (BCPA) was developed [21]. Cuckoo Search (CS) was first used in 2009 [38]. Dolphin Partner Optimization (DPO) was first implemented in 2009 [33]. The Firefly Algorithm (FA) was first used in 2010 [39]. In 2012, the Bird Mating Optimizer (BMO) was introduced [3]. In 2012, the Krill Herd (KH) was discovered [9]. In 2012, the Fruit Fly Optimization Algorithm (FOA) was developed [27]. In 2006, the Group Search Optimizer (GSO) was introduced [12, 13], in 2014, the A. Grey wolf optimizer [22], in 2015, the Moth-Flame Optimization (MFO) algorithm [23], and in 2016, the Whale Optimization Algorithm (WOA).

The notions of evolution in nature are often used as inspiration in EAs. The GA algorithm is the most widely used algorithm in this field. Holland invented this method in 1992 [35], and it is based on Darwinian evolution ideas and replicates them. EAs seek for the global optimum in a search space by generating one or more random solutions for a given issue [36], which are then evaluated. This collection of solutions is referred to as the set of candidate solutions. The pool of candidates is then refined repeatedly until it meets the requirements of a termination condition. A more precise estimate of the global optimal than the initial random guesses might be deemed to constitute the increase in performance. This method provides evolutionary algorithms with a number of fundamental benefits, including problem independence, derivation independence, avoidance of local optima, simplicity, and derivation independence. In general, optimization is accomplished by the evolution of an initial random solution in evolutionary algorithms. In each new generation, a new population is formed by the combination and mutation of the individuals from the preceding generation. Because the best people have a greater likelihood of participating in the formation of the new population, the new population is more likely to be superior to the preceding generation in quality (s). This has the potential to ensure that the original random population is optimized over the course of many generations of evolution. Differential evolution (DE) [8], Evolutionary Programming (EP) [11, 30], and Evolution Strategy (ES) [17, 18], Genetic Programming (GP) [32], Biogeography-Based Optimizer (BBO) [6], and Teaching learning based optimization (TLBO) [19] are examples of EAs (TLBO). An example of this would be the BBO algorithm, which was initially suggested in 2008 [19] by Simon. Biological geography, which is the study of biological species in terms of their geographic distribution, served as the inspiration for the fundamental concept of this method (over time and space). There might be many case studies covering various islands, regions, and even continents over the course of decades, centuries, or millennia. It is the goal of this area of research to explore the relationships between various species (inhabitants) in terms of immigration, emigration, and mutation by investigating diverse ecosystems (habitats or territories). The fundamental idea for the BBO algorithm came from the development of ecosystems (taking into account various sorts of species such as predator and prey) through time via migration and mutation to achieve a stable state.

It is the notion of issues as "black boxes" that leads to the independencies between problems and derivations. Evolutionary algorithms only make use of the issue formulation in the context of assessing a collection of potential solution alternatives. The basic process of optimization is carried out fully independently of the issue and on the basis of the inputs and outputs that have been supplied by the user. As a result, the nature of the issue is not a worry while using evolutionary algorithms, but the representation is a critical stage in the process of using them. This is the same reason why evolutionary algorithms do not need the derivation of the issue in order to find the global optimum in their solutions. Another benefit of evolutionary algorithms is that they have a high degree of local optima avoidance because of their stochastic character. An evolutionary algorithm stuck in a local optimality is subjected to a stochastic operator, which results in a random change in the solution and, ultimately, an escape from the local optimality. Although there is no certainty that this problem will be totally resolved, stochastic algorithms have a far better possibility of escaping from local optima when compared to deterministic approaches. It is also not guaranteed that an evolutionary algorithm would provide an extremely exact approximation of the global optimum, but by repeating the process numerous times, the possibility of discovering a better solution increases. Last but not least, the simplicity of evolutionary algorithms is another trait that distinguishes them. Natural evolutionary notions or social behaviors serve as the primary sources of inspiration for the vast majority of algorithms in this subject, despite the fact that they are quite straightforward. Furthermore, evolutionary algorithms adhere to a broad and universal structure, in which a collection of randomly generated solutions is repeatedly improved or developed. The way of improving this set is what distinguishes algorithms in this subject from other approaches.

The motivation of this paper to get new algorithm combining between the advantage of Gravitational search algorithm and the advantage of differential evolution algorithm (DE) to balancing between the diversity and the convergence rate for solving the global optimization problem.

The paper is organized as follows. Section 2 introduces the GSA, whereas Section 3 introduces the DE. Section 4 presents the proposed GSA-DE, while Section 5 provides a full theoretical and experimental study. Section 6 discusses the findings of the experiments before concluding.

2 Gravtional Search Algorithm

Newton's laws of motion and gravity are used in GSA. It is used in different applications [29]. Nevertheless, the algorithm is yet unknown in the study. In GSA, each mass (agent) has four specifications: inertial, location, active and passive gravity masses. The mass location is (b) to enhance bat-algorithm behavior for higher dimensional situations. And (c) to prevent trapping into local optima by increasing population variety. The low algorithm intensity relative to the diversity is also a drawback.

The GSA's physics may be defined using the definitions above. for $i = 1, \ldots, n$ let the *ith* particle's location in the D-dimensional search space be, $P_i^t(p_{i1}^t, p_{i2}^t, \ldots, p_{id}^t)$ The attraction force (F_{ijD}^t) between the ith agent and the jth agent is defined as follows:

$$F_{\text{ijD}}^t = G^t \times \frac{M_{\text{pi}}^t \times M_{\text{aj}}^t}{R_{ij}^t} \times (p_{id}^t - p_{\text{jd}}^t) \tag{2.1}$$

The passive gravitational mass associated with the ith agent at time t, M_{pi}^t.the active gravitational mass associated with the *jth* particle at time *t*, G^t.is the gravitational constant at the current time, and, R_{ij}^t is the Euclidean distance between the two particles i and j at the current time are all defined as $d = 1, \ldots, D$ and the Euclidean distance as following equation.

$$R_{ij}^t = \left\| P_i^t, P_j^t \right\|^2 \tag{2.2}$$

The formula for determining the gravitational constant, G^t, is as follows:

$$G^t = G^{t0} \times e^{-\alpha \frac{\text{iteration}}{\text{max iteration}}} \tag{2.3}$$

where α and G^{t0} are descending coefficient and initial value ,respectively.

Ith agent's total force of attraction at each time t in a D-dimensional space is represented by the equation:

$$F_{id}^t = \sum_{i=1\ ,i\neq j}^{\text{ng}} \text{rand} \times F_{\text{ijd}}^t \tag{2.4}$$

Agent number ng is the uniform random number generator $(rand)$ in [0,1]. This introduces the algorithm's stochastic character. The ith particle's acceleration is determined by the equation:

$$ac_{id}^t = \frac{F_{id}^t}{M_{ii}^t} \tag{2.5}$$

Where M_{ii}^tt is the inertial mass of the i^{th} agent .The velocity and position of particles are calculated as follows:

$$V_{id}^{t+1} = rand \times V_{id}^t + \text{ac}_{id}^t \tag{2.6}$$

$$x_{id}^{t+1} = x_{id}^t + V_{id}^{t+1} \tag{2.7}$$

To generate random numbers in the range [0,1], use rand. To determine the gravitational and inertial masses, the fitness tests are used. If the particle has more mass, it might be considered a better particle and therefore have a greater degree of attraction, which can impact other particles with a high level of attraction. The following equations will be used to update the gravitational and inertial mass:

$$M_{ai} = M_{\text{pi}} = M_{ii} \qquad for\ i = 1, \ldots, ng \tag{2.8}$$

$$m_i^t = \frac{\text{fitness}_i^t - \text{best}^t}{\text{best}^t - \text{worst}^t} \tag{2.9}$$

$$M_i^t = \frac{m_i^t}{\sum_{i=1}^{\text{ng}} m_i^t} \tag{2.10}$$

where fitness_i^t represents the fitness value of the ith particle at time t. And $\text{best}^t, \text{worst}^t$ is the best and worst agent in the population.

The procedure of GSA is explained in Algorithm 1.

Algorithm 1. Gravitational Search Algorithm.

```
Initialization and generation and evaluation initial population
T=1;
While t< maximum of iterations;
Evaluate each particle's fitness.
Recalculate the mass of the particles in motion.
Update the acceleration of the particle.
Update the velocity of the particles.
Mutate and update the locations of particles.
T= t +1;
End while;
```

3 Differential evaluation algorithm

D-dimensional parameter vectors (NG D-dimensional parameter vectors) are used in the DE search, which is population-based stochastic search [35].

$$P_i^t(p_{i1}^t,\ \ p_{i2}^t, \ldots, p_{id}^t)\ for\ \ i = 1, \ldots, n$$

Experimental vectors are generated in the present population by the application of mutations and crossings for each generation. Once the election process is complete, it is determined which vector will be passed down to the next generation of creatures.

When random elements in the population are perturbed to discover a better solution, it is called a "mutation." One or more vector differences may be used by DE to make mutants known as donor vectors $\vec{V}_{i,G}$. And here's a brand-new answer thanks to the equation:

$$\vec{V}_{i,G} = P_a^t + F \times \left(P_b^t - P_c^t\right) \tag{3.1}$$

a,b,c are three random solutions from the population, and F is a random vector generated by using a uniform distribution in [0,1].

Differential evolution develops a new solution $\vec{U}_{i,G}$. through crossover operation to increase the population's diversity. This is the simplest way to generalize the crossover operation of Differential evolution:

$$\vec{U}_{i,G} = \begin{cases} \vec{V}_{i,G} & if\ rand < CR \ \ or \ \ (j = j_0) \\ P_i^t & Otherwise \end{cases} \tag{3.2}$$

The [0, 1] range is represented by the random number generator's notation. To regulate the percentage of variables that are copied from the donor vector, the CR crossover rate is a real number between 0 and 1.

Finally, the new approach will be put to the test. This is the first step, which is termed the process of success, and it involves comparing the new solution's goal to the present solution's target function.

$$P_i^t = \begin{cases} \vec{U}_{i,G} & if\ \ f\left(\vec{U}_{i,G}\right) \le f(P_i^t) \\ P_i^t & Otherwise \end{cases} \tag{3.3}$$

Algorithm 2. Differential evolution Algorithm.

Initialization and generation and evaluation initial population
T=1;
While t< maximum of iterations;
Evaluate each particle's fitness
for i = 1 to NP do
Randomly select $a \neq b \neq c$ *from the set* $I = \{1, 2, \ldots, NP\}$.
Generate a donor vector $\vec{V}_{i,G} = P_a^t + F \times \left(P_b^t - P_c^t\right)$
Generate new Solution by using crossover $(\vec{V}_{i,G},\ P_i^t)$
If $\left(\vec{U}_{i,G}\right) \le f\left(P_i^t\right)$
$p_i^t = \vec{U}_{i,G}$
Else
$p_i^t = P_i^t$;
End if
End For
T=t+1;
End while;

4 Gravtional evolution algorithm (GEA)

Global optimization issues have been solved using a novel technique called Gravitational evolution algorithm (GEA). Gravitational search and differential evolution are now combined in a new algorithm that relies on the behavior of the new algorithms to achieve its goals.

Building the algorithm was motivated by the goal of creating a new algorithm that can solve global optimization issues by integrating two fundamental methods.

It starts with establishing an initial population and then evaluating its fitness before creating a new population utilizing Gravitational search method stages. Finally, random numbers are generated by employing uniform distribution in the interval [0,1].

Afterwards, we perform a random number comparison with the control parameter and use the differential evolution algorithm to generate new populations from the current population; if the number is greater than the control parameter, we update the control parameter in order to help the algorithm that uses the control parameter to achieve its goal. To prevent algorithms from appearing in local optimums, use the following procure to describe the algorithm step:

Algorithm 3. Pseudo-code of Gravitational Evolution Algorithm.

Initialization and generation and evaluation initial population
T=1;
While t< maximum of iterations;
Evaluate each particle's fitness
Update the particle mass ();
Particle acceleration update ();
Particle velocity update ();
Mutate and update the locations of particles
Generating random number r by using uniform distribution
If $r > \beta$
Generate new population by using DE
Else
$\beta = \lambda\ \beta$
End If
T=t+1;
End while;

5 Performance evaluation

5.1 Benchmark problem and comparative algorithms

The CEC2014 [19] benchmark tasks include 10 and 30 variables, respectively, and are used to assess GEA. Additional algorithms that are used as comparisons include the GSA, DE/rand/1, DE/best/1 [35], GWO [22], and WOA [24]. Each algorithm's parameters are exactly the same as the original paper's settings. The CEC2014 benchmark suit's optimization problems may be divided into four types: unimodal functions (F1-F3), simple multimodal functions (F4-F16), hybrid functions (F17-F22), and composition functions (F17-F22). All four types of optimization problems are included in the benchmark suit (F23-F30).

Table 1: Result of comparative between GEA and other 5 algorithms in cec2014 benchmark problem in 10 dimensional

Problem	*Statistics*	*GEA*	*GSA*	*DE/best/1*	*DE/rand/1*	*WOA*	*GWO*
F_1	*Mean*	*3.85E+02*	*463869.2*	*1106812*	*32606.1*	*3890473*	*8.15E+06*
	STD	*5.52E+02*	*879712.5*	*4719003*	*34246.62*	*3553463*	*5.11E+06*
	Best	*3.01E-01*	*6387.598*	*3.919661*	*949.2715*	*343649.8*	*193264.6*
	p-value		*8.86E-05*	*0.247145*	*8.86E-05*	*8.86E-05*	*8.86E-05*
	H		+	+	+	+	+
F_2	*Mean*	*0.663629*	*1511.576*	*497.6204*	*2824.918*	*104654.6*	*13311771*
	STD	*0.86589*	*1657.083*	*2203.1*	*3431.44*	*72028.82*	*59520266*
	Best	*0.00158*	*6.177039*	*0.003318*	*0.217371*	*9563.852*	*247.981*
	p-value		*8.86E-05*	*0.006425*	*8.86E-05*	*8.86E-05*	*8.86E-05*
	H		+	+	+	+	+
F_3	*Mean*	*8.48E-02*	*16881.1*	*1421.934*	*133.9318*	*31919.05*	*5.13E+03*
	1.37E-01	*6452.203*	*3579.571*	*264.9373*	*20346.63*	*3.90E+03*	
	Best	*3.07E-05*	*4665.932*	*4.37E-05*	*4.16E-05*	*5242.159*	*598.5873*
	p-value		*8.86E-05*	*0.000103*	*3.90E-04*	*8.86E-05*	*8.86E-05*
	H		+	+	+	+	+
F_4	*Mean*	*25.38853*	*32.73411*	*22.88489*	*27.58266*	*21.7603*	*29.62865*
	STD	*15.63968*	*10.46249*	*16.65576*	*13.31403*	*24.7378*	*15.63753*
	Best	*0.014752*	*0.131943*	*0.000485*	*0.454419*	*1.013873*	*6.210831*
	p-value		*0.007189*	*0.116888*	*0.370261*	*0.455273*	*0.20433*
	H		+	=	-	+	+
F_5	*Mean*	*1.90E+01*	*16.99962*	*2.02E+01*	*20.17736*	*20.07501*	*2.03E+01*
	STD	*4.47E+00*	*7.33E+00*	*7.27E-02*	*0.059387*	*0.06418*	*1.12E-01*
	Best	*1.52E-09*	*3.14E-09*	*20.02695*	*20.07036*	*20.00778*	*20.057*
	p-value		*1.56E-01*	*8.86E-05*	*8.86E-05*	*8.86E-05*	*8.86E-05*
	H		+	+	+	+	+
F_6	*Mean*	*2.078706*	*3.222534*	*2.861052*	*0.32762*	*6.943859*	*1.459603*
	STD	*1.921019*	*1.422662*	*1.657056*	*0.58719*	*1.902649*	*0.944804*
	Best	*6.30E-09*	*1.27E-07*	*0.303479*	*0*	*4.272517*	*0.464084*
	p-value		*0.025094*	*0.108427*	*0.000681*	*0.00012*	*0.295878*
	H		+	+	-	+	+
F_7	*Mean*	*3.52E-02*	*0.014283*	*2.152793*	*0.035744*	*0.939322*	*9.62E-01*
	STD	*2.43E-02*	*0.014301*	*8.716378*	*0.054179*	*0.579672*	*7.67E-01*
	Best	*0.00E+00*	*0 0.022151*	*0*	*0.264915*	*0.072006*	
	p-value		*3.59E-03*	*0.000219*	*3.32E-01*	*8.86E-05*	*8.86E-05*
	H		-	+	+	+	+
F_8	*Mean*	*16.31729*	*23.63023*	*12.74981*	*1.695541*	*42.21536*	*12.89865*
	STD	*10.75675*	*7.806794*	*4.706167*	*2.106128*	*12.52399*	*6.641443*
	Best	*4.974795*	*10.94454*	*2.984877*	*0*	*22.89269*	*3.985396*
	p-value		*0.020633*	*0.16711*	*8.86E-05*	*0.000254*	*0.411465*
	H		+	+	-	+	+
F_9	*Mean*	*1.79E+01*	*22.93375*	*15.19005*	*9.273538*	*42.1206*	*1.57E+01*
	STD	*7.15E+00*	*9.767079*	*4.675335*	*4.895965*	*16.22071*	*8.12E+00*
	Best	*8.95E+00*	*10.94454*	*7.959667*	*0.994959*	*20.03892*	*4.976026*
	p-value		*6.45E-02*	*0.167147*	*5.93E-04*	*8.86E-05*	*2.18E-01*
	H		+	+	-	+	+
F_{10}	*Mean*	*613.9967*	*822.8948*	*296.3796*	*67.46152*	*462.0718*	*332.6139*
	STD	*216.8621*	*253.8105*	*171.1789*	*47.44162*	*227.8012*	*158.8374*
	Best	*342.4704*	*457.6187*	*7.079739*	*27.23255*	*21.85193*	*61.14805*
	p-value		*0.052222*	*0.000593*	*8.86E-05*	*0.056915*	*0.000189*
	H		+	-	-	+	-

Continued on next page

Table 1 – Continued from previous page

problem	Statistics	GEA	GSA	DE/best/1	DE/rand/1	WOA	GWO
F_{11}	Mean	7.01E+02	855.9786	341.0562	374.8357	966.4483	3.70E+02
	STD	2.89E+02	271.7638	227.0596	233.9324	295.3582	2.30E+02
	Best	2.46E+02	350.8849	26.88665	18.65945	314.5692	15.26307
	p-value		5.69E-02	0.001162	4.49E-04	1.69E-02	1.32E-03
	H		+	-	-	+	-
F_{12}	Mean	0.014386	4.75E-09	0.203613	0.487978	0.684653	0.749078
	STD	0.018264	9.83E-10	0.198809	0.163472	0.276335	0.554753
	Best	1.38E-09	2.82E-09	0.037463	0.132112	0.2463	0.022385
	p-value		0.061953	8.86E-05	8.86E-05	8.86E-05	8.86E-05
	H		+	+	+	+	+
F_{13}	Mean	9.02E-02	0.043607	0.203499	0.132899	0.437103	1.61E-01
	STD	3.94E-02	0.022913	0.054227	0.033255	0.175125	5.73E-02
	Best	1.95E-02	0.013099	0.115449	0.073057	0.182339	0.072148
	p-value		8.92E-04	8.86E-05	3.59E-03	8.86E-05	1.71E-03
	H		-	+	+	+	+
F_{14}	Mean	0.292428	0.468725	0.25276	0.168248	0.305449	0.267501
	STD	0.08811	0.025568	0.131733	0.068658	0.19155	0.206424
	Best	0.107322	0.398221	0.094256	0.084795	0.090195	0.064516
	p-value		8.86E-05	1.67E-01	0.000681	0.736875	3.70E-01
	H		+	=	-	+	+
F_{15}	Mean	9.37E-01	1.168924	1.725484	1.686619	7.38372	1.70E+00
	STD	1.89E-01	0.404312	2.626247	0.234089	3.531207	9.87E-01
	Best	6.90E-01	0.601433	0.277532	1.29579	1.886543	0.216893
	p-value		7.31E-02	0.390533	8.86E-05	8.86E-05	8.97E-03
	H		+	+	+	+	+
F_{16}	Mean	3.045887	3.906425	2.555709	2.276467	3.18786	2.550387
	STD	0.498835	0.228766	0.35986	0.386315	0.349926	0.52636
	Best	1.853671	3.533899	1.821683	1.517101	2.618621	1.270299
	p-value		0.00014	0.002821	0.000338	0.2322267	0.012374
	H		+	-	-	+	-
F_{17}	Mean	7.03E+02	230510.5	858.10267	2983.075	79187.567	3.36E+04
	STD	5.02E+02	134394.1	605.7743	5014.965	119085.4	1.25E+05
	Best	1.03E+02	41659.25	12.66251	58.55799	1832.235	1094.91
	p-value		8.86E-05	0.331723	4.55E-01	8.86E-05	1.03E-04
	H		+	+	+	+	+
F_{18}	Mean	1949.28	7795.5	3014.055	5027.793	11045.69	7655.239
	STD	1459.8	6504.794	3219.054	8177.103	11957.66	6896.576
	Best	62.79862	1313.06	70.3685	2.294568	112.3256	342.2171
	p-value		0.000163	0.501591	0.331723	0.016881	0.002495
	H		+	+	+	+	+
F_{19}	Mean	2.61E+00	2.875193	3.066475	0.852347	5.389349	2.59E+00
	STD	1.13E+00	0.977112	1.523437	0.623679	1.298181	9.76E-01
	Best	1.43E+00	1.560957	0.305825	0.026828	3.318074	1.223434
	p-value		3.13E-01	0.108427	2.93E-04	8.86E-05	9.70E-01
	H		+	+	-	+	+
F_{20}	Mean	28.47036	10949.5	1972.176	1230.428	5486.235	4023.204
	STD	15.61367	9294.014	1562.594	2339.985	4358.713	3400.187
	Best	4.401841	1436.47	29.48037	0.009246	67.70872	98.63171
	p-value		8.86E-05	0.000103	0.007189	8.86E-05	8.86E-05
	H		+	+	+	+	+
F_{21}	Mean	2.88E+02	7740.598	179644.1	261.2517	19806.86	6.39E+03
	STD	1.55E+02	2958.713	788583.4	633.2085	18243.08	4.65E+03
	Best	8.10E-01	3424.554	0.336544	0.37314	1689.866	218.1373
	p-value		8.86E-05	0.765198	7.31E-02	8.86E-05	1.03E-04
	H		+	+	-	+	+
F_{22}	Mean	166.995	219.1806	135.2963	10.50327	70.07111	97.25276
	STD	125.4289	77.7014	119.1683	31.49959	53.67533	58.48716
	Best	20.33763	135.5189	8.495923	8.63E-05	23.76806	21.42924
	p-value		0.135357	0.411465	8.86E-05	0.012374	0.100458
	H		+	+	-	-	+
F_{23}	Mean	3.29E+02	290.6202	329.4575	329.4575	310.8089	3.33E+02
	STD	1.75E-13	60.86603	2.32E-12	1.75E-13	47.76279	5.11E+00
	Best	3.29E+02	200	329.4575	329.4575	200	329.459
	p-value		3.13E-02	0.000244	1.00E+00	7.31E-02	8.86E-05
	H		-	+	=	+	+

Continued on next page

Table 1 – Continued from previous page

problem	Statistics	GEA	GSA	DE/best/1	DE/rand/1	WOA	GWO
	Mean	142.8404	199.6945	142.0038	118.601	175.3054	147.6158
	STD	26.941	8.824695	31.51826	20.41209	24.92134	35.54399
F_{24}	Best	115.7576	162.6706	111.0595	106.7013	125.0573	114.452
	p-value		0.00014	0.736875	0.001162	0.001944	0.654159
	H		+	+	-	+	+
	Mean	1.98E+02	199.1443	194.7637	186.7806	190.9998	1.98E+02
	STD	4.23E+00	1.780228	18.30307	30.26616	14.02922	7.20E+00
F_{25}	Best	1.89E+02	192.0321	142	112.1846	164.7425	175.533
	p-value		9.11E-01	0.247145	9.70E-01	4.79E-02	7.37E-01
	H		+	+	+	=	+
	Mean	100.0949	126.1873	100.2166	100.1478	100.3463	105.1335
	STD	0.039607	42.0694	0.133601	0.030311	0.186161	22.32929
F_{26}	Best	100.0184	100.023	100.0966	100.0935	100.1722	100.069
	p-value		0.001944	0.000189	0.002204	8.86E-05	0.003185
	H		+	+	+	+	+
	Mean	2.56E+02	359.2686	355.0081	288.5103	309.9264	3.38E+02
	STD	1.56E+02	120.708 124.3	104.5875	179.5254	8.56E+01	
F_{27}	Best	8.53E-01	4.025149	2.901943	1.62175	6.613538	5.654857
	p-value		1.24E-02	0.008968	8.52E-01	1.17E-01	5.22E-02
	H		+	+	+	+	+
	Mean	552.6827	942.8052	524.1853	388.7098	562.5016	454.4154
	STD	119.7546	374.3374	101.2692	24.41232	119.6309	64.02529
F_{28}	Best	372.3805	442.6589	380.8277	368.8503	383.4287	368.9162
	p-value		0.000189	0.20433	0.000219	0.97022	0.003185
	H		+	+	-	+	-
	Mean	6.17E+05	813502.2	659955.8	189955.1	1414.783	6.79E+05
	STD	1.29E+06	2761047	1042441	582018.6	1720.433	9.54E+05
F_{29}	Best	2.42E+02	200.0432	292.6526	275.8483	451.2893	288.4881
	p-value		7.31E-02	0.411465	4.55E-01	2.47E-01	1.67E-01
	H		+	+	+	+	+
	Mean	1718.427	2508.615	1285.303	498.4708	1413.566	1264.045
	STD	541.3451	395.6997	552.5531	34.23181	502.7726	633.1467
F_{30}	Best	818.7443	1910.478	521.2681	459.6285	905.9334	640.2725
	p-value		8.86E-05	0.022769	8.86E-05	0.092963	0.018675
	H		+	=	-	+	=
+/ =/-			27/ 0/3	24/ 3/3	15/ 1/14	28/ 1/1	25/ 1/4

5.2 Parameter setting

The population size for all algorithms 20 and the number of function evaluation 5000. the results obtained over 30 simulation runs on 10 and 30 dimensional problems. The parameter of algorithms in the following table :

Table 2: Parameter setting for comparative algorithms

Algorithm	Parameter
GEA	$G_0 = 100$, $\alpha = 20$, $CR = 0.5$, $\beta_0 = 0.95$ $\lambda = 0.01$
GSA [29]	$G_0 = 100$, $\alpha = 20$,
DE/best/1	$CR = 0.5$,
DE/rand/1	$CR = 0.5$,

Table 3: Result of comparative between GEA and other 5 algorithms in cec2014 benchmark problem in 30 dimensional

Problem	Statistics	GEA	GSA	DE/best/1	DE/rand/1	WOA	GWO
	Mean	667627.8	6237614	1684684	40841834	67949082	89518032
	STD	563181.1	7987349	1093151	18232274	36405413	73100381
F_1	Best	127891.1	1309033	308883	14434354	21173958	9614474
	p-value		8.86E-05	0.001507	8.86E-05	8.86E-05	8.86E-05
	H		+	+	+	+	+

Continued on next page

Table 3 – Continued from previous page

problem	Statistics	GEA	GSA	DE/best/1	DE/rand/1	WOA	GWO
F_2	Mean	0.001308	11058.18	7.91E+08	7862.934	1.72E+08	4.5E+09
	STD	0.003504	6849.6	2.93E+09	10878.39	80545169	3.56E+09
	Best	2.99E-07	2974.755	1.10E-07	0.044694	58161744	2.8E+08
	p-value		8.86E-05	0.016881	8.86E-05	8.86E-05	8.86E-05
	H		+	+	+	+	+
F_3	Mean	2.52E+01	10144.89	3049.66	978.4523	62492.37	3.51E+04
	STD	7.16E+01	6824.924	11578.22	2596.34	31496.26	8.66E+03
	Best	1.51E-01	3064.247	4.12E-03	8.15E-04	27403.6	22053.16
	p-value		8.86E-05	0.82276	3.04E-02	8.86E-05	8.86E-05
	H		+	+	+	+	+
F_4	Mean	109.95	137.4525	155.2774	73.75978	258.1728	283.4716
	STD	74.9823	57.00341	176.7873	33.49893	72.13673	94.15227
	Best	0.000328	62.91509	72.40274	3.944113	93.64468	143.0072
	p-value		0.370261	0.575486	0.085924	8.86E-05	0.000254
	H		+	+	=	+	+
F_5	Mean	2.00E+01	19.99943	2.09E+01	20.91175	20.48712	2.10E+01
	STD	4.02E-05	7.54E-04	6.73E-02	0.05758	0.109876	8.04E-02
	Best	2.00E+01	2.00E+01	20.75556	20.79444	20.29949	20.78932
	p-value		1.03E-04	8.86E-05	8.86E-05	8.86E-05	8.86E-05
	H		-	+	+	+	+
F_6	Mean	14.12128	17.69086	16.53511	6.263875	36.15582	16.09415
	STD	3.224036	3.513745	2.226101	6.144979	3.82999	2.063827
	Best	9.66E+00	1.15E+01	12.57151	0.784998	29.67258	12.7874
	p-value		0.001944	0.016881	0.001944	8.86E-05	0.061953
	H		+	+	-	+	+
F_7	Mean	6.40E-03	0.000986	1.53842	0.00074	2.29165	3.04E+01
	STD	7.38E-03	0.003137	3.42429	0.002276	0.754443	2.42E+01
	Best	1.14E-13	1.14E-13	2.03E-08	1.14E-13	1.172086	3.821705
	p-value		2.54E-02	0.00014	4.88E-03	8.86E-05	8.86E-05
	H		+	+	-	+	+
F_8	Mean	94.96847	141.2836	77.500057 51.62525	194.0107	96.10558	
	STD	22.34932	19.63808	18.28073	21.0799	34.16896	16.89391
	Best	61.68736	99.49566	53.7277	5.969772	145.0126	76.32618
	p-value		0.000103	0.008968	1.40E-04	8.86E-05	0.575486
	H		+	-	-	+	+
F_9	Mean	1.04E+02	159.9886	87.90236	176.3461	250.3255	1.13E+02
	STD	1.89E+01	20.39191	20.5653	10.16481	50.93287	1.77E+01
	Best	7.16E+01	119.3946	45.76806	162.7904	173.6467	76.95122
	p-value		8.86E-05	0.025094	8.86E-05	8.86E-05	1.67E-01
	H		+	-	+	+	+
F_{10}	Mean	2832.649	3263.908	1581.316	955.1262	4580.143	2394.668
	STD	676.7583	496.5074	457.1767	500.7177	820.325	567.1696
	Best	1809.324	2416.296	838.4113	34.43011	3454.864	1402.165
	p-value		0.056915	0.00014	1.03E-04	0.000103	0.125859
	H		+	-	-	+	+
F_{11}	Mean	3.33E+03	3642.088	2813.785	6612.077	5540.301	2.94E+03
	STD	7.45E+02	682.8017	1192.513	330.7355	1092.479	7.99E+02
	Best	1.90E+03	2441.607	1263.659	5968.36	4058.298	1537.54
	p-value		2.79E-01	0.052222	8.86E-05	8.86E-05	2.32E-01
	H		+	=	+	+	+
F_{12}	Mean	0.015133	4.17E-03	2.090195	2.215647	1.979992	1.518394
	STD	0.010377	2.87E-03	0.322131	0.290287	0.480952	1.444104
	Best	3.92E-03	3.41E-04	1.481114	1.703721	1.179206	0.071706
	p-value		8.86E-05	8.86E-05	8.86E-05	8.86E-05	8.86E-05
	H		-	+	+	+	+
F_{13}	Mean	4.47E-01	0.278702	0.524991	0.390819	0.577794	5.89E-01
	STD	1.11E-01	0.079026	0.139065	0.044452	0.114017	4.22E-01
	Best	2.80E-01	0.164599	0.278518	0.288831	0.376517	0.308242
	p-value		3.38E-04	6.74E-02	7.31E-02	1.32E-03	3.70E-01
	H		-	+	+	+	+
F_{14}	Mean	0.285315	0.291366	3.426109	0.486107	0.269097	9.886986
	STD	0.051464	0.059272	10.26294	0.205593	0.041448	14.45088
	Best	0.181788	0.1829	0.209979	0.225442	0.204717	0.185635
	p-value		7.37E-01	1.71E-03	0.001713	0.217957	1.94E-03
	H		+	+	+	+	+

Continued on next page

Table 3 – Continued from previous page

problem	Statistics	GEA	GSA	DE/best/1	DE/rand/1	WOA	GWO
F_{15}	Mean	3.77E+00	3.392776	74.75662	16.59824	81.24459	2.65E+02
	STD	8.24E-01	0.466447	68.00072	1.11895	24.27719	5.75E+02
	Best	2.76E+00	2.616831	13.21259	14.63464	27.16183	6.579439
	p-value		1.00E-01	8.86E-05	8.86E-05	8.86E-05	8.86E-05
	H		+	+	+	+	+
F_{16}	Mean	12.60043	13.56877	11.5602	12.7341	12.81127	11.65923
	STD	0.639314	0.454536	0.540247	0.244046	0.535354	0.53497
	Best	11.44622	12.6254	10.17982	12.32336	11.66388	10.50938
	p-value		0.00012	0.000293	0.370261	0.247145	0.001019
	H		+	-	+	+	-
F_{17}	Mean	2.14E+05	437368.2	416759	1603947	8167213	2.62E+06
	STD	1.75E+05	306026.3	829178.8	830035	5329622	2.50E+06
	Best	4.27E+04	82570.6	36482.56	519222.8	1207766	205632
	p-value		3.19E-03	0.940481	8.86E-05	8.86E-05	8.86E-05
	H		+	+	+	+	+
F_{18}	Mean	43484.83	914.7734	64369.31	2731.222	16075.35	15395195
	STD	176119.9	772.1582	261851.5	4205.749	12907.87	25781415
	Best	91.27151	192.6129	126.5056	71.85256	1536.526	1054.713
	p-value		0.011129	0.910825	0.247145	0.005111	0.001507
	H		=	+	+	+	+
F_{19}	Mean	2.71E+01	61.74118	23.03501	7.935345	60.70276	5.26E+01
	STD	2.70E+01	30.17722	21.90087	1.306813	46.59386	3.00E+01
	Best	7.47E+00	12.54163	8.236851	6.610922	21.67442	13.03359
	p-value		4.55E-03	0.525653	3.90E-04	5.73E-03	2.28E-02
	H		+	+	-	+	+
F_{20}	Mean	5660.579	30074.82	721.4711	2855.205	52897.56	21310.66
	STD	6272.075	9271.806	762.3886	2340.64	35267.76	10962.08
	Best	743.8283	18119.01	218.2495	586.572	8992.888	8398.697
	p-value		1.03E-04	0.000189	0.030365	8.86E-05	8.92E-04
	H		+	-	-	+	+
F_{21}	Mean	3.68E+04	180005	42881.64	289741.3	2975122	1.55E+06
	STD	3.56E+04	101263.7	35646.96	246128.5	2896806	2.56E+06
	Best	6.75E+03	48352.5	5095.031	66279.93	258855.6	126458
	p-value		1.63E-04	0.501591	8.86E-05	8.86E-05	8.86E-05
	H		+	+	+	+	+
F_{22}	Mean	724.3076	994.9896	466.1796	224.1792	773.8759	365.5827
	STD	197.3793	244.8511	260.9867	141.8209	255.7901	150.5611
	Best	299.8226	592.5554	22.44461	3.66E+01	332.782	163.4354
	p-value		0.006425	0.005734	1.03E-04	0.681322	0.000254
	H		+	-	-	-	+
F_{23}	Mean	3.16E+02	317.7301	318.8699	315.2441	332.4548	3.41E+02
	STD	3.62E-01	1.554507	6.63E+00	1.20E-09	46.44225	1.62E+01
	Best	3.15E+02	315.4597	315.2441	315.2441	200	323.6844
	p-value		1.03E-04	0.178956	8.86E-05	1.37E-02	8.86E-05
	H		+	+	-	+	+
F_{24}	Mean	229.9085	202.3923	250.1204	228.9086	205.437	200.0049
	STD	7.760945	6.640863	4.794322	5.304299	3.990782	0.002684
	Best	221.4875	200.1777	243.1871	223.0381	200.5205	200.0022
	p-value		8.86E-05	8.86E-05	0.575486	8.86E-05	8.86E-05
	H		-	+	+	-	-
F_{25}	Mean	2.16E+02	203.175	208.6594	212.51	206.5557	2.12E+02
	STD	4.08E+00	4.20437	3.881211	4.41234	11.80739	6.56E+00
	Best	2.10E+02	200 204.1836	206.166	200	200	
	p-value		8.86E-05	0.000449	3.66E-02	2.20E-03	1.52E-02
	H		-	-	=	+	+
F_{26}	Mean	165.2026	200.0332	111.2348	110.5036	105.4485	180.1364
	STD	48.72286	0.012524	30.44398	45.18269	22.255527	40.83523
	Best	100.2377	200.0162	100.4497	100.2886	100.2494	100.4172
	p-value		0.135357	0.016881	0.002821	8.92E-04	0.82276
	H		+	-	-	-	+
F_{27}	Mean	6.54E+02	1389.785	633.6296	434.1811	1168.279	6.36E+02
	STD	1.93E+02	807.1348	222.2086	74.34489	315.909	1.54E+02
	Best	4.01E+02	403.2864	402.0386	300	436.5972	425.0887
	p-value		1.94E-03	0.97022	8.92E-04	3.38E-04	8.23E-01
	H		+	+	-	+	+
F_{28}	Mean	2391.824	3739.562	1362.61	873.1818	2325.205	1408.449

Continued on next page

Table 3 – Continued from previous page

problem	Statistics	GEA	GSA	DE/best/1	DE/rand/1	WOA	GWO
	STD	851.4217	1084.972	357.2145	45.55553	746.7455	381.4609
	Best	1171.148	1914.937	909.0175	804.8997	200	960.5843
	p-value		0.000892	0.00039	8.86E-05	0.940481	0.00014
	H		+	-	-	+	-
	Mean	5.15E+06	5425.771	2294129	1576351	6584511	2.37E+06
	STD	1.27E+07	1711.749	4203688	3206355	5539303	5.90E+06
F_{29}	Best	1.06E+03	2795.072	1448.382	1397.66	19230.08	4431.213
	p-value		7.31E-02	0.085924	2.04E-01	7.31E-02	7.31E-02
	H		+	+	+	+	+
	Mean	23354.46	8896.285	23445.46	4737.893	167573.9	73249.14
	STD	37823.8	4969.024	40837.24	1561.962	94872.32	42881.62
F_{30}	Best	2033.09	4328.352	2153.431	3471.821	44943.22	18499.75
	p-value		4.55E-01	0.575486	2.06E-02	0.00012	0.003592
	H		+	+	+	+	+
+/=/-			24/1/5	20/1/9	17/2/11	27/0/3	25/0/5

5.3 Evaluation metric

We use five metric for comparative between after solve each benchmark function we will compute the error rate between fitness function of algorithms and optimal solution for each benchmark problem and then we compute the three statistical measurement (mean ,standard deviation and best (minimum value)) also we use Wilcoxon sign rank test for comparing between GEA and other comparative algorithms if the sign of test (H) is "+" that mean outperforms GEA on other algorithm and if sign "-" that mean outperforms other algorithm on GEA while no different between GEA and other algorithm if the sign of test is "=". Also we compute the p-value for Wilcoxon test if the p-value is near of zero that mean different between GEA and other algorithm and no different if the p-value near of 1.

6 Result and discussion result

6.1 Result of comparative between the algorithms

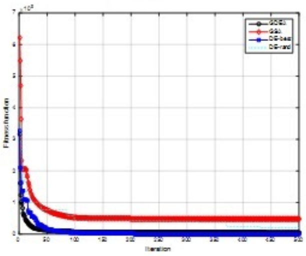

Figure 1: Benchmark function

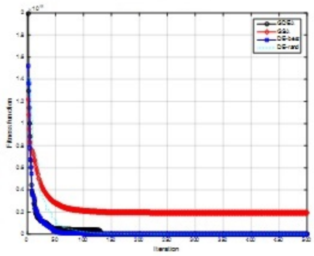

Figure 2: Benchmark function

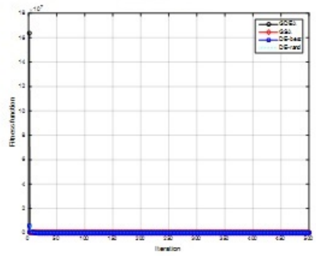

Figure 3: Benchmark function

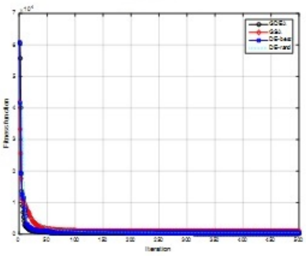

Figure 4: Benchmark function

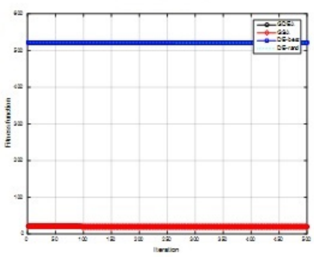

Figure 5: Benchmark function

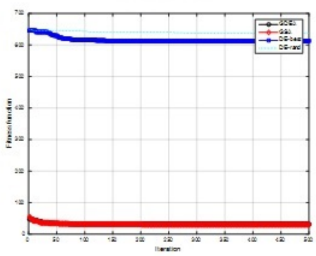

Figure 6: Benchmark function

6.2 Discussion Result

For 10 dimensional problems table 2 shown outperforms GEA on all other algorithms in function

$$(F1, F2, F3, F6, F12, F15, F17, F18, F20, F25, F26, F27, F29)$$

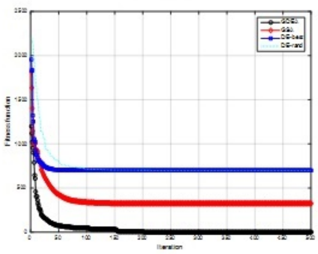

Figure 7: Benchmark function

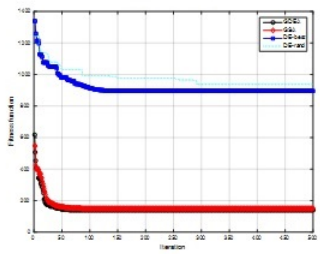

Figure 8: Benchmark function

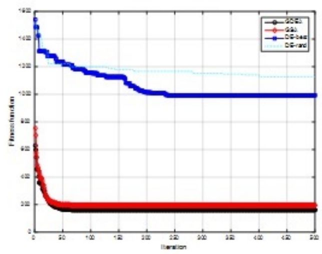

Figure 9: Benchmark function

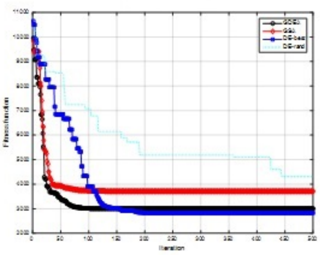

Figure 10: Benchmark function

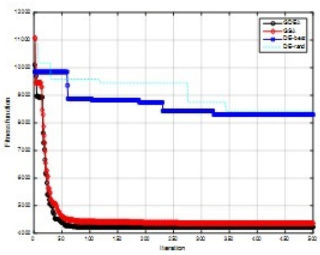

Figure 11: Benchmark function

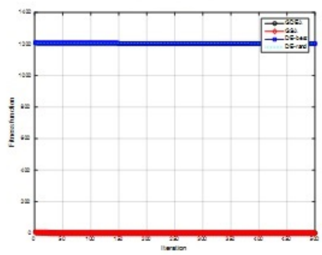

Figure 12: Benchmark function

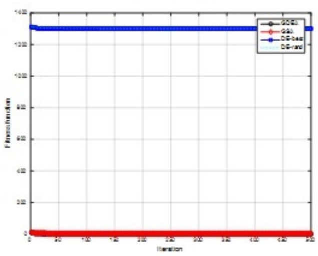

Figure 13: Benchmark function

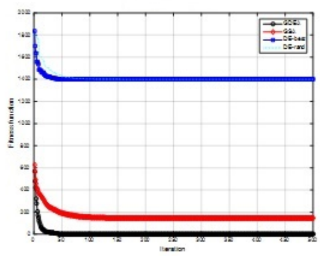

Figure 14: Benchmark function

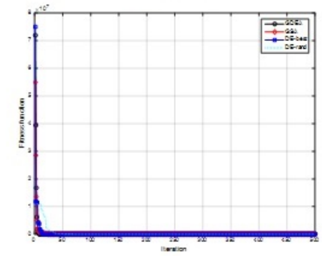

Figure 15: Benchmark function

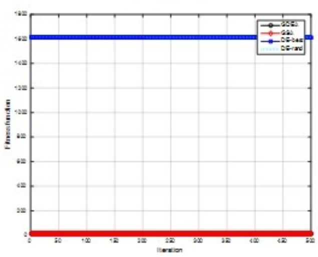

Figure 16: Benchmark function

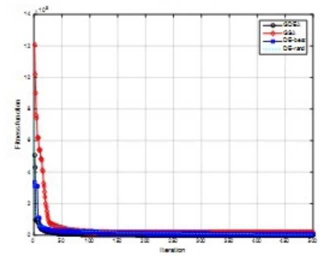

Figure 17: Benchmark function

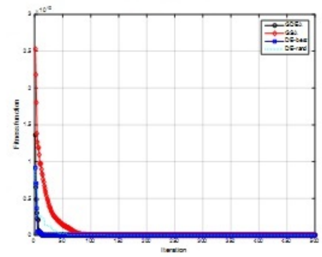

Figure 18: Benchmark function

and we can see outperforms GEA on original GSA in 27 functions ,and outperform DE/best/1 in 24 function and outperforms DE/rand/1 algorithm in 15 functions and with comparative with WOA, Table 2 shown superior GEA in 28 functions and 25 functions with GWO.

For 10 dimensional problems, Table 2 shows outperforms GEA on all other algorithms in functions (F1, F2, F3, F6, F12, F15, F17, F18, F20, F25, F26, F27, F29) and we can see that it outperforms GEA on the original GSA in 27 functions, and outperforms DE/best/1 in 24 functions and outperforms DE/rand/1 algorithm in 15 functions. With comparative with WOA, Table 2, it shows superior GEA in 28 functions and 25 functions with GWO.
Overall the final rank for algorithms in 10-dimensional problems:

$$\text{GEA} \longrightarrow DE/rand/1 \ \longrightarrow DE/best/1 \longrightarrow GWO \longrightarrow WOA$$

In 30-dimensional problems Table 3 outperforms GEA on original GSA in 24 functions, and outperforms DE/best/1 in 20 function and outperforms DE/rand/1 algorithm in 17 functions and with comparative with WOA table 2 shown

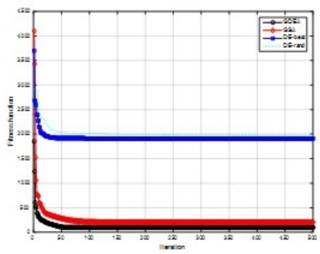

Figure 19: Benchmark function

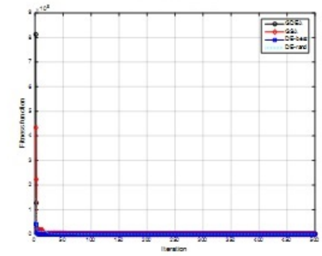

Figure 20: Benchmark function

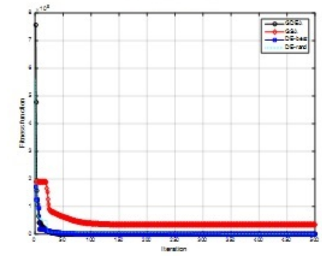

Figure 21: Benchmark function

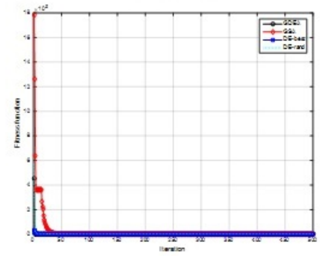

Figure 22: Benchmark function

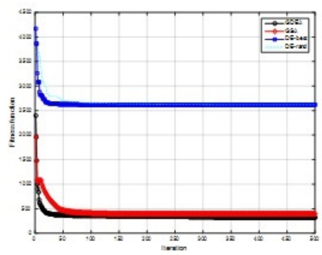

Figure 23: Benchmark function

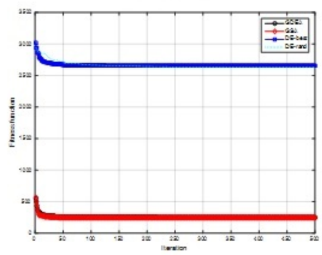

Figure 24: Benchmark function

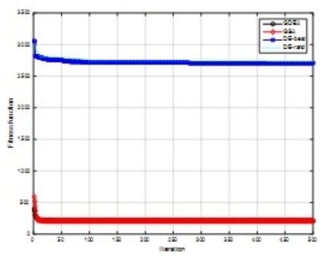

Figure 25: Benchmark function

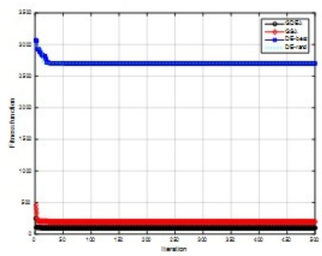

Figure 26: Benchmark function

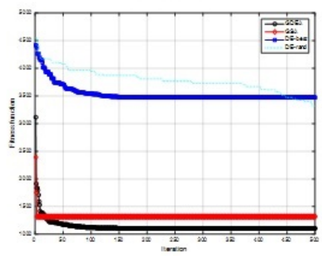

Figure 27: Benchmark function

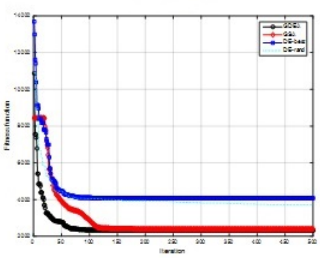

Figure 28: Benchmark function

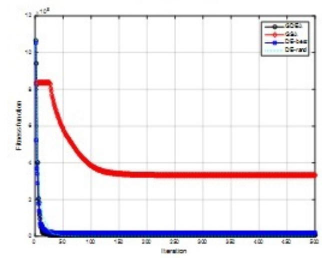

Figure 29: Benchmark function

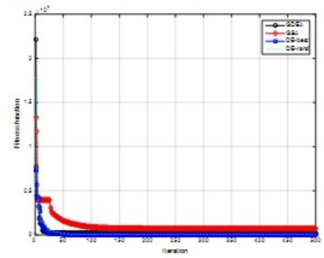

Figure 30: Benchmark function

superior GEA in 27 functions and 25 functions with GWO.
Finally GEA algorithm shown the outperforms on other algorithms when increasing the dimensional of variable we getting the best result comparing with others algorithms.

7 Conclusion

GSA and DE are combined to develop a new hybrid algorithm in this paper. Combining GSA's convergence capabilities with DE's variety is the primary goal of this project. For the purpose of validating the GEA's performance against the standard GSA, thirty benchmark functions are employed, as well as two methods for DE and WOA, as well as GWO, which use CEC2014. In most functions minimizing, GEA surpasses all other algorithms, according to the findings. GEA's exploitation speed is also shown to be quicker than that of other algorithms.

References

[1] H.A. Abbass, *Marriage in honey bees optimization – a haplometrosis polygynous swarming approach*, Evol. Comput. Proc. 2001 Congress, 2001, p. 207–214.

[2] B. Alatas, *Artificial chemical reaction optimization algorithm for global optimization*, Expert Syst. Appl. **38** (2011), 13170–13180.

[3] A. Askarzadeh and A. Rezazadeh, *A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer*, Int. J. Energy Res. **37** (2013), no. 10, 1196–1204.

[4] M. Dorigo, M. Birattari and T. Stutzle, *Ant colony optimization*, IEEE Comput. Intell. Magaz. **1** (2006), no. 4, 28–39.

[5] O.K. Erol and I. Eksin, *A new optimization method: big bang–big crunch*, Adv. Eng. Softw. **37** (2006), 106–111.

[6] A.V. Fiacco and G.P. McCormick, *Nonlinear programming: Sequential unconstrained minimization techniques*, New York: Wiley, 1968.

[7] R.A. Formato, *Central force optimization: a new metaheuristic with applications in applied electromagnetics*, Prog. Electromag. Res. **77**, (2007) 425–491.

[8] D. Fogel, *Artificial intelligence through simulated evolution*, Wiley-IEEE Press, 2009.

[9] A.H. Gandomi, A.H. Alavi, H. Krill, *Krill herd: A new bio-inspired optimization algorithm*, Commun. Nonlinear Sci. Numer. Simul. **17** (2012), no. 12, 4831–4845.

[10] A. Hatamlou, *Black hole: a new heuristic optimization approach for data clustering*, Inf. Sci. **222** (2013), 175–184.

[11] N. Hansen, S.D. Müller and P. Koumoutsakos, *Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMAES)*, Evol. Compu. **11** (2003), 1–18.

[12] S. He, S.Q.H. Wu and J.R. Saunders, *A novel group search optimizer inspired by animal behavioural ecology*, IEEE Int. Conf. Evol. Comput. 2006, p 1272–1278.

[13] S. He, Q. Henry Wu and J.R. Saunders, *Group search optimizer: an optimization algorithm inspired by animal searching behavior*, IEEE Trans. Evol. Comput. **13** (2009), no. 5, 973–990.

[14] A. Kaveh and M. Khayatazad, *A new meta-heuristic method: ray optimization*, Comput. Struct. **112** (2012), 283–94.

[15] A. Kaveh and S. Talatahari, *A novel heuristic optimization method: charged system search*, Acta Mech. **213** (2010), 267–89.

[16] J. Kennedy and R. Eberhart, *Particle swarm optimization, in Neural Networks*, Proc. IEEE Int. Conf. 1995, p. 1942–1948.

[17] J.R. Koza, *Genetic programming*, 1992.

[18] D. Simon, *Biogeography-based optimization*, Evol. Comput. IEEE Trans. **12** (2008), 702–13.

[19] J.J. Liang, B.Y. Qu and P.N. Suganthan, *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization*, Comput. Intell. Lab. Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore **635** (2013), 490.

[20] X. Li, *A new intelligent optimization-artificial fish swarm algorithm*, Ph.D Thesis, Zhejiang University of Zhejiang, China, 2003.

[21] X. Lu and Y. Zhou, *A novel global convergence algorithm: bee collecting pollen algorithm*, Int. Conf. Intell. Comput. Springer, Berlin, Heidelberg, 2008, p. 518–525.

[22] S.A. Mirjalili, S.M. Mirjalili and A. Lewis, *Grey wolf optimizer*, Adv. Engin. Software **69** (2014), 46–61.

[23] S.A. Mirjalili, *Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm*, Knowledge-Based Syst. **89** (2015), 228–249.

[24] S.A. Mirjalili and A. Lewis, *The whale optimization algorithm*, Adv. Engin. Software **95** (2016), 51–67.

[25] F.F. Moghaddam, R.F. Moghaddam and M. Cheriet, *Curved space optimization: a random search based on general relativity theory*, arXiv, preprint arXiv, 1208. 2214 2012.

[26] A. Mucherino and O. Seref, *A novel metaheuristic search for global optimization*, Proc. Conf. Data Min. Syst. Anal. Optim. Biomed. 2007, p. 28–30.

[27] W-T. Pan , *A new fruit fly optimization algorithm: taking the financial distress model as an example*, Knowl-Based Syst. **26** (2012), 69–74.

[28] P.C. Pinto, T.A. Runkler and J.M. Sousa, *Wasp swarm algorithm for dynamic MAX-SAT problems*, Int. Conf. Adapt. Natural Comput. Algor. Springer, Berlin, Heidelberg, 2007, p. 350–357.

[29] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, *a gravitational search algorithm*, Inf. Sci. **179** (2009), 2232–2248.

[30] G. Rechenberg, *Evolution strategy*, Comput. Intell. Imitat. Life **1** (1994).

[31] M. Roth and W. Stephen *Termite, A swarm intelligent routing algorithm for mobile wireless ad-hoc networks*, Stigmergic Optim. 2005, p 155–184.

[32] H.-P. Schwefel, *Evolution and optimum seeking*, New York: Wiley, 1995.

[33] Y. Shiqin, J. Jianjun and Y. Guangxing, *A dolphin partner optimization*, IEEE WRI Glob. Cong. Intell. Syst. **1** (2009), 124–128.

[34] H. Shah-Hosseini, *Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation*, Int. J. Comput. Sci. Eng. **6** (2011), 132–40.

[35] R. Storn and K. Price, *Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces*, J. Glob. Optim. **11** (1997), 341–59.

[36] X. Yao, Y. Liu and G. Lin, *Evolutionary programming made faster. Evolut Comput*, IEEE Trans. **3** (1999), 82–102.

[37] X-S. Yang, *A new metaheuristic bat-inspired algorithm*, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), ed., Springer, 2010, p. 65–74.

[38] X-S. Yang and S. Deb, *Cuckoo search via Lévy flights*, Nature Biologically Inspired Computing, NaBIC, World Congress, 2009, p. 210–14.

[39] X-S. Yang, *Firefly algorithm, stochastic test functions and design optimisation*, Int. J. Bio-Inspired Comput. **2** (2010), 78–84.