Levy Flight and Chaos Theory-Based Gravitational Search Algorithm for Global Optimization: LCGSA for Global Optimization

Sajad Ahmad Rather, Pondicherry University, India*

https://orcid.org/0000-0002-0351-7705

P. Shanthi Bala, Pondicherry University, India

https://orcid.org/0000-0003-0576-4424

ABSTRACT

The gravitational search algorithm (GSA) is one of the highly regarded population-based algorithms. It has been reported that GSA has a powerful global exploration capability but suffers from the limitations of getting stuck in local optima and slow convergence speed. In order to resolve the aforementioned issues, a modified version of GSA has been proposed based on Levy flight distribution and chaotic maps (LCGSA). In LCGSA, the diversification is performed by utilizing the high step size value of Levy flight distribution while exploitation is carried out by chaotic maps. The LCGSA is tested on 23 well-known classical benchmark functions. Moreover, it is also applied to three constrained engineering design problems. Furthermore, the analysis of results is performed through various performance metrics like statistical measures, convergence rate, and so on. Also, a signed Wilcoxon rank-sum test has been conducted. The simulation results indicate that LCGSA provides better results as compared to standard GSA and most of the competing algorithms.

KEYWORDS

Chaotic Maps, Engineering Design Optimization, Global Optimization, Gravitational Search Algorithm (GSA), Hybridization, LCGSA, Levy Flight, Swarm Intelligence

1. INTRODUCTION

Optimization is the process of finding the best solution to the problem from the already available pool of solutions. Basically, a problem whether in engineering, business, science, or management can be considered as an optimization task. To solve an optimization problem, the researchers have used both deterministic methods and population-based algorithms. In the former case, the optimization algorithm depends on the gradient information of the problem. However, they have the drawbacks of unimodality, entrapment in local minima, and so on. In contrast, population-based algorithms treat an optimization problem as a black box meaning non-dependability on the design of the given problem. Moreover, they have the advantages of stochasticity, simplicity of design, flexibility, information interchange, parallelism, and less complexity.
All population-based algorithms are heuristic in nature which means finding the optimal solution from the feasible candidate solutions. Basically, the optimization process starts with the initialization of searcher agents in the solution space. Then, the iteration process creates rapid changes in the values of candidate solutions. After the maximum number of iterations, the optimization process stops and provides an optimal solution. Furthermore, it has been seen that all population-based algorithms consist of two fundamental processes namely exploration and exploitation. The exploration (diversification) is defined as the lower and upper limits of the solution space where searcher agents can move during the optimization process. Moreover, searcher agents change values more often during the exploration phase. In contrast, exploitation (intensification) is the process of finding an optimal solution from the rich pool of feasible candidate solutions. Meanwhile, the searcher agents undergo less number of changes during this phase. According to Eiben and Schippers (1998), exploration and exploitation are inversely proportional to each other. So, a proper balance between them during the optimization process is necessary for getting the best solutions.

It is a fact that most of the population-based algorithms are inspired by nature. The researchers have proposed a number of heuristic algorithms to solve real-world problems. Examples of some of the famous HAs include GA (Holland, 1992), PSO (Kennedy and Eberhart, 1995), ACO (Dorigo et al., 2006), DE (Storn and Price, 1995), BBO (Simon, 2008), and many more. Moreover, the recent inclusions into the list of HAs include GWO (Mirjalili et al., 2014), ALO (Mirjalili, 2015), SCA (Mirjalili, 2016), SSA (Mirjalili et al., 2017), AFO (Cheng et al., 2018), SRO (Shabani et al., 2019), BWO (Hayyolalam, 2020), CSO (Ahmed et al., 2020), and BMO (Suliman et al., 2020).

In the hierarchy of HAs, Gravitational Search Algorithm (GSA) is one of the efficient, highly used, and popular optimization technique designed by Rashedi et al. (2009). It is a physics-based algorithm inspired by Newton’s laws of universal gravitation and motion. In GSA, the searcher agents are in the form of masses which get attracted towards heavy masses. The position of heavy mass provides the optimal solution. It has been reported that GSA has a powerful global exploration capability which is crucial for solving complex optimization problems.

In the work of Yin et al. (2011), GSA has been combined with k-harmonic means to perform clustering. Moreover, Mirjalili et al. (2012) have trained MLP by employing GSA and PSO for function approximation and classification. The GSA has also been applied to image processing (Chakraborti et al., 2014), speech processing (Prajna et al., 2014), power scheduling (Gouthamkumar et al., 2015), load dispatch (Güvenç et al., 2012), and many other applications. However, it has been reported in a number of studies (Rather et al., 2020a, 2020b; Mirjalili, 2017) that GSA suffers from the problems of getting stuck in local optima and slow convergence speed. To overcome the aforementioned drawbacks, a modified version of GSA based on levy flight and chaos theory has been proposed. We have named it LCGSA.

The main contributions of the paper are summarized as follows:

- New hybrid heuristic technique namely LCGSA for global optimization has been proposed.
- In LCGSA, intensification is carried out by ten different chaotic maps, and exploration power is provided by levy flight distribution.
- For performance evaluation, 23 classical benchmark functions and three engineering design problems have been employed.
- Besides, statistical test and nine state of the art algorithms have been used for comparative analysis.

The remaining sections of the paper are organized as follows: Section 2 deals with the motivation behind the study. The literature survey of different HAs for global optimization is dealt in Section 3. A brief description of the standard GSA is introduced in Section 4. Likewise, Section 5 covers the proposed LCGSA algorithm. Moreover, the experimental results of benchmark functions are presented.
in Section 6. Besides, simulation results of LCGSA on the engineering benchmarks are discussed in Section 7. Furthermore, Section 8 provides the conclusion and future directions of the paper.

2. MOTIVATION

The levy flight distribution is a very useful applied mathematics technique. It is basically a power series having infinite variance and variable step size. These properties are essential while dealing with sensitivity in initialization and falling in local minima problems of optimizations techniques.

Also, it has been reported in a number of studies that non-linear chaotic progression(s) are superior as compared to random numbers (Gandomi et al., 2013). In fact, any unsymmetrical change in the chaotic sequences create huge ripples in the chaotic system. Moreover, chaotic sequences can take candidate solutions from infeasible regions of search spaces towards the feasible global minima neighborhood.

In addition, it is the first time; levy flight and chaos theory techniques have been hybridized to solve global optimization and engineering design problems. Previously, researchers have used chaotic maps to tackle simple function optimization problems (Shen et al., 2015; Basset et al., 2019). Besides, the aforementioned studies used only a few chaotic maps, and no proper statistical analysis of simulation results were provided. In contrast, the current study uses levy flight and ten distinct chaotic maps embedded with GSA for mathematical and engineering optimization.

Basically, LCGSA has both high exploitation and strong global exploration capabilities due to its hybrid nature. The above properties are pivotal in resolving slow convergence and exploitation limitations of standard GSA. Mathematically speaking, the diverse standard functions from the classical benchmark suite and engineering problems will rigorously test the global optima finding capability and fractional constraint handling power of LCGSA.

3. LITERATURE SURVEY

Gravitational Search Algorithm (GSA) is a very promising optimization technique that is inspired by physics. It is based on the principle of mass attraction. The candidate solutions are actually masses. Obviously, a heavy mass object is having a high-intensity field and attracts other feasible candidate solutions towards itself. Consequently, the position of the heavy mass object gives the best solution. Further, GSA is having better exploration power but it has the issues of premature convergence speed and entrapment in local minima. Researchers have provided a number of innovative approaches to fix GSA problems. One of the solutions is provided by Rodenas et al. (2019). They utilized chaotic maps and quasi-Newton methods to provide equilibrium balance between diversification and intensification phases in standard GSA. The modified version of GSA was named memetic GSA. It has been applied to a number of real world test functions and showed promising outcomes.

In standard GSA, the heavy masses are important for getting global optima as they have high fitness function values. However, heavy masses can also create neighborhood local minima problems which results in premature convergence. To alleviate this problem, an improved version of GSA was developed to increase the exploitation capability of GSA. To test the efficiency of GSA, it was applied to Loney’s solenoid design problem (Khan et al., 2020).

It has been reported that PSO has a high speed of convergence but it has the drawback of reduced diversity of candidate solutions. Consequently, GSA was hybridized with PSO to provide the global searching capability. The hybrid PSO-GSA model exhibited efficient convergence speed as compared to other participating algorithms (Jiang et al., 2020).

The global exploration ability of GSA has been utilized to solve a number of problems including feature selection. In fact, the binary quantum version of GSA has been combined with a k-nearest neighbor classifier to increase the classification accuracy and reduce the dimensions of the recognition datasets (Barani et al., 2017).
The fuzzy logic concepts have been combined with GSA to optimally design IIR (Infinite Impulsive Response) filter. The fuzzy GSA provided better robustness and stability as compared to standard GSA and DE (Pelusi et al., 2018).

Most recently, researchers have utilized the concept of species niche from environmental sciences for solving function optimization and real-world problems through mathematical modelling. Basically, Haghbayan et al. (2017) have proposed niche GSA in order to optimize multimodal functions. In niche GSA, the swarm niches are formed using the nearest neighbor classifier whereas the hill valley algorithm detects population niches. The simulation results depicted high performance of niche GSA.

In another work, Yazdani et al. (2014) modified GSA by making improvements in mass equations and used elitism criterion for finding multiple local minima in benchmark functions.

In the last decade, several quality articles have been written on the topic of object tracking. It is a trending topic in the field of computer vision due to its potential applications in vehicle navigation, traffic monitoring, etc. Likewise, deep learning concepts have been combined with GSA to track objects optimally in video frames (Kang et al., 2018).

It is a fact that GSA has an efficient global searching capability. There were number of attempts made by researchers to enhance the exploration proficiency of GSA. Actually, Khajooei et al. (2016) have introduced the concepts of positive and negative masses to advance the diversification power of standard GSA. Similarly, a discrete version of GSA was developed to solve the 0-1 knapsack problem by modifying the position equation and fitness function of standard GSA. The simulation outcomes indicated better performance of discrete GSA in terms of faster convergence rate and better accuracy (Sajedi et al., 2017).

Quite recently, Zandevakil et al. (2019) have come up with a modified GSA based on the notion of attractive and repulsive forces (AR-GSA) to overcome premature convergence problems of standard GSA. Moreover, AR-GSA was applied to CEC-2013 test functions and exhibited promising simulation results.

It has been seen that chaos theory has a number of applications in many areas of computer science, meteorology, environmental sciences, and so on. Actually, chaos theory deals with the study of dynamical systems whose disorder states are highly reactive to any perturbation in the seed values. Moreover, chaos theory is based on the principle of the butterfly effect. Besides, Mirjalili et al. (2017) have combined chaotic maps with the gravitational constant parameter of standard GSA in order to provide symmetry between diversification and intensification stages. The situation outcomes showed better performance of chaotic GSA as compared to standard GSA.

Rather et al. (2020b) have applied chaotic GSA for engineering optimization. They solved three mechanical engineering design problems using chaotic GSA. Besides, it was shown that CGSA provided better results as compared to competing algorithms including classical GSA.

Similarly, chaotic maps have been combined with GSA to identify parameters of the chaotic systems. It was accomplished by utilizing chaotic maps for local search to increase the speed of convergence while GSA carried out global exploration of the solution space (Li et al., 2012).

In another research, the stochastic and ergodicity properties of chaotic maps have been embedded with GSA to solve eight unconstrained numerical test functions. It has been seen that CGSA provided better performance as compared to GSA and different PSO versions (Gao et al., 2014).

It is a fact that PSO is having powerful exploitation capability but it suffers from the drawback of reduced diversification proficiency. The aforementioned problem of PSO has been solved by using chaotic sequences in place of random numbers. Actually, eight chaotic maps were embedded with classical PSO to increase its exploration power. The simulation results confirmed the efficiency of chaotic PSO versions (Alatas et al., 2009). Moreover, Alatas (2010) has developed a chaotic version of the Artificial Bee Colony (ABC) algorithm to overcome premature convergence and local minima entrapment problems of the traditional ABC algorithm. In addition, Alatas (2010) has also designed chaotic mathematical models of Harmony Search (HS) algorithms to resolve convergence issues of standard HS algorithm. Besides, seven chaotic maps including sinusoidal, tent, heneon, and so on, have
been utilized to provide chaotic sequences for the initialization of the search space. The experimental results depicted the optimal performance of chaotic HS algorithms.

Differential Evolution (DE) is one of the famous evolutionary algorithms utilized for solving optimization problems. However, DE has the drawbacks of local minima falling and premature convergence. To overcome these disadvantages, self-adaptive chaotic maps were introduced in traditional DE (Zhenyu et al., 2006).

Simulated Annealing (SA) is another famous heuristic technique mainly utilized for solving combinatorial optimization tasks. It has a simple design, efficient diversification power but slow convergence speed. The chaotic maps were employed to provide chaotic initialization in place of Gaussian distribution to overcome local searching problems of SA (Mingjun et al., 2004). Recently, the Firefly Algorithm (FA) has also been combined with chaos theory for robust global optimization. In fact, 12 chaotic maps were used for FA parameter tuning (Gandomi et al., 2013).

In the family of HAs, BBO is a newly invented stochastic technique utilized mainly for solving real-world optimization problems due to its novel algorithmic design structure. However, while dealing with complex problems, BBO faces the problems of local minima entrapment and premature convergence. Roughly speaking, ten chaotic maps were utilized to overcome the aforementioned disadvantages of BBO. Besides, experimental studies confirmed the optimal performance of chaotic BBO (Saremi et al., 2014).

Likewise, 13 chaotic maps were used to increase the global searching capability of the BA algorithm (Gandomi et al., 2014). Moreover, Cuckoo Search (CS) has also been hybridized with chaos theory to improve its solution quality and boost convergence speed (Wang et al., 2015).

In the last decade, researchers have used levy flight distribution to solve a number of real-world problems in different academic fields such as cryptography, economics, data science, engineering, and many more. Basically, levy flight is a heavy-tailed Markov process with variable step length. It has the characteristics of randomness, infinite variance, stable distribution, and scale-invariant property. The aforementioned properties of levy distribution are pivotal for the resolution of premature convergence and diversity issues of HAs. In fact, a levy flight trajectory with a mutation strategy has been introduced in BA to overcome its convergence and local minima entrapment problems. The benchmark function outcomes demonstrated the efficient performance of levy-based BA as compared to classical BA (Xie et al., 2013). In another work, levy flight and opposition based learning were combined with BA, respectively, to resolve its slow convergence and exploration issues. Moreover, 16 test functions were utilized for checking the efficiency of the modified BA (Shan et al., 2016). Besides, Li et al. (2019) have designed yet another improved version of BA using levy distribution and exponentially decreasing inertia weight function. The effectiveness of the modified BA was tested on various benchmark functions and two engineering design problems.

In order to overcome early convergence and global searching drawbacks of PSO, levy distribution was combined with the position equation of PSO. Actually, the variable step size of levy random walk helped PSO particles to get away from local minima traps. Moreover, widely used test functions were utilized for the performance evaluation of chaotic PSO (Hakli et al., 2014). Besides, levy flight distribution has been hybridized with PSO to update its velocity equation. Consequently, it results in faster convergence and improved global exploration of the particles towards feasible regions of the search space (Jenki et al., 2016).

It has been seen that the Fish Swarm Algorithm (FSA) has appreciable exploration capability but it has the problem of slow convergence rate. So, levy flight random walk and dynamic firefly factor were introduced in FSA to enhance its exploitation power. The simulation results of benchmark functions indicate improvement in the solution quality and overall accuracy of the classical FSA (Peng et al., 2018).

Multi-Verse Optimizer (MVO) (Mirjalili et al., 2016) is one of the recent HAs inspired by the concepts of astrophysics. In MVO, exploration is performed by white and black holes while intensification is carried out by worm holes. Besides, searcher agents are in the form of the universe(s).
It has been observed that MVO has the disadvantages of entrapment in local minima and low diversity of candidate solutions. To overcome these shortcomings, MVO was hybridized with levy flight in order to deal with complex and non-linear search spaces. Moreover, modified levy MVO was applied to numerical and test scheduling problem(s) to demonstrate its efficiency and real world problem-solving capability (Hu et al., 2016).

The summarization of the related works is shown in Table 1. It clearly indicates that both chaos theory and levy flight have been rigorously used to solve real-world and domain-specific problems. Besides, randomness, ergodicity, and stochasticity properties of chaos theory and levy flight distribution have been utilized to solve premature convergence, entrapment in local minima, and accuracy issues of HAs. Thus, in this study, LCGSA is applied to three engineering design problems to demonstrate its capability in handling non-linear, complex, and unknown search spaces.

4. STANDARD GRAVITATIONAL SEARCH ALGORITHM (GSA)

GSA is one of the highly regarded physics-based HA. It is inspired by the law of gravitation and motion. In fact, gravity is one of the four basic forces in nature. The other three forces are weak nuclear force, electromagnetic force, and strong nuclear force. Moreover, the law of gravitation is basically an inverse square law which states that “the attractive force between two masses is directly proportional to the product of their masses and inversely proportional to the square of the distance between them” (Halliday et al., 2000; Rashedi et al., 2009; Rather et al., 2019a, 2019b, 2019c, 2019d, 2021a, 2021b).

The GSA is first initialized with a random distribution of searcher agents in the form of masses. The force between the point masses is calculated in Equation (1).

$$F_{ij} = G(t) \frac{m_{pi}(t) m_{aj}(t)}{R_{ij}(t) + \epsilon} (x^d_j(t) + x^d_i(t))$$

where $m_{pi}(t)$ and $m_{aj}(t)$ are passive and active gravitational masses, respectively. The Euclidian distance is represented as $R_{ij}(t)$ while $\epsilon$ is a small constant.

To get a proper balance between exploration and exploitation, the GSA utilizes an important parameter called gravitational constant represented by ‘G’. Besides, it helps in the accuracy of the search. It is given by Equation (2).

$$G(t) = G(t_0) e^{-\frac{CI}{MI}}$$

where $G(t)$ and $G(t_0)$ are the values of the gravitational constant at time interval $t$ and $t_0$, respectively. Also, $\alpha$ is an exponentially decreasing coefficient whereas CI and MI correspond to the current iteration and the maximum number of iteration(s).

As the masses are moving in the search space and each of them is exerting a force. So, the total force is given by Equation (3).

$$F_i^d(t) = \sum_{j=1,j\neq i}^{m} \gamma_j F_{ij}$$

where $\gamma_j$ has values between 0 and 1.
### Table 1. Outline of the Related Research Works associated with GSA, Chaotic Maps, and Levy Flight

| Authors (Year) | Technique | Chaotic Maps / Levy Flight | Objective of Optimization |
|----------------|-----------|----------------------------|---------------------------|
| Jiang et al. (2020) | GSA/PSO | No | Enhancement in exploration |
| Khan et al. (2020) | GSA | No | To solve electromagnetic design problem |
| Rather et al. (2020b) | GSA | Chaotic maps | Engineering design optimization |
| Rodenas al. (2019) | GSA | Chaotic maps | To increase solution quality and accuracy |
| Zandevakili et al. (2019) | GSA | No | Function optimization |
| Li et al. (2019) | BA | Levy flight | Benchmark function and engineering optimization |
| Huang et al. (2019) | FFA | Levy flight | Breast cancer detection |
| Pelusi et al. (2018) | Fuzzy GSA | No | IIR filter design |
| Kang et al. (2018) | GSA | No | Object tracking |
| Amirsadri et al. (2018) | GWO | Levy flight | MLP training |
| Peng et al. (2018) | FSA | Levy flight | To increase accuracy |
| Barani et al. (2017) | GSA | No | Classification |
| Mirjalili et al. (2017) | GSA | Chaotic maps | Function optimization |
| Sajedi et al. (2017) | Discrete GSA | No | To solve knapsack problem |
| Haghbayan et al. (2017) | Niche GSA | No | Function optimization |
| Khajoei et al. (2016) | GSA | No | GSA performance enhancement |
| Shan et al. (2016) | BA | Levy flight | Improvement in BA |
| Li et al. (2016) | MFO | Levy flight | To increase convergence rate of MFO |
| Jensi et al. (2016) | PSO | Levy flight | Robust global optimization |
| Hu et al. (2016) | MVO | Levy flight | Test scheduling |
| Wang et al. (2015) | CS | Chaotic maps | Improvement in solution quality |
| Yazdani et al. (2014) | GSA | No | Multimodal optimization |
| Saremi et al. (2014) | BBO | Chaotic maps | To overcome local minima problem |
| Hakili et al. (2014) | PSO | Levy flight | To increase exploration power of PSO |
| Gandomi et al. (2014) | BA | Chaotic maps | To advance diversification capability of BA |
| Xie et al. (2013) | BA | Levy flight | Global optimization |
| Gandomi et al. (2013) | FA | Chaotic maps | Global optimization |
| Li et al. (2012) | GSA | Chaotic maps | Chaotic system parameter identification |
| Gao et al. (2012) | GSA | Chaotic maps | Unconstrained function optimization |
| Alatas (2010) | ABC | Chaotic maps | Numerical optimization |
| Alatas (2010) | HS | Chaotic maps | To solve benchmark functions |
| Alatas et al. (2009) | PSO | Chaotic maps | Function optimization |
| Zhenyu et al. (2006) | DE | Chaotic maps | To resolve problems of DE algorithm |
| Mingjun et al. (2004) | SA | Chaotic maps | To increase agent diversity |
Furthermore, after a number of iterations, the heavy masses will be scattered throughout the search space which represents feasible solutions. So, it is important to preserve the quality of the best solutions. Therefore, the elitism criterion i.e. kbest strategy is used in GSA. It means that only optimal and efficient heavy masses will execute force in all directions after the fulfillment of the stopping criterion. It is shown in Equation (4).

\[ F^d_i(t) = \sum_{j \neq k_{\text{best}}, j \neq i} \gamma_j F^d_{ij}(t) \]  

(4)

Moreover, the acceleration of the masses is calculated according to the second law of motion as given in Equation (5).

\[ a^d_i(t) = \frac{F^d_i(t)}{m_i(t)} \]  

(5)

In GSA, the point masses get attracted to heavy masses because they have the highest intensity and strong force of attraction. Hence, the position and velocity of the heavy mass are pivotal for finding the global optimum which is provided in Equations (6) and (7), respectively.

\[ v^d_i(t + 1) = \gamma_j v^d_i(t) + a^d_i(t) \]  

(6)

\[ x^d_i(t + 1) = x^d_i(t) + v^d_i(t + 1) \]  

(7)

The step-wise description of GSA is presented in Algorithm 1.

Algorithm 1: Standard Gravitational Search Algorithm (GSA)
1: Randomized initialization of the search space
2: Calculate the fitness of each mass
3: Initialize the parameters including the maximum number of iterations \(T\), initial value of the gravitational constant \(G(t_0)\), and coefficient \(\alpha\)
4: Start the iteration counter at \(t=0\)
5: \textbf{while} \(t < T\) \textbf{do}
6: \hspace{1em} \textbf{for} each candidate solution \textbf{do}
7: \hspace{2em} Update the gravitational constant, \(G(t)\)
8: \hspace{2em} Find the gravitational force, \(F^d_i(t)\) using Equation (4)
9: \hspace{2em} Calculate the mass acceleration, \(a^d_i(t)\) by using Equation (5)
10: \hspace{2em} Update the mass velocity, \(v^d_i(t + 1)\) with the help of Equation (6)
11: \hspace{2em} Update the mass position, \(x^d_i(t + 1)\) using Equation (7)
12: \hspace{1em} \textbf{end for}
13: \hspace{1em} \(t = t + 1\)
14: \hspace{1em} \textbf{end while}
15: Return the optimal candidate solution
5. LEVY FLIGHT AND CHAOS THEORY BASED GRAVITATIONAL SEARCH ALGORITHM (LCGSA)

In this section, first, the preliminary concepts of levy flight and chaos theory are introduced to provide a firm foundation for the proposed LCGSA algorithm. In fact, levy flight is employed to provide exploration capability and proper balance between exploration and exploitation in LCGSA. Furthermore, 10 chaotic maps will help in the local search and speed up the convergence of the candidate solutions towards the global optima.

5.1. Levy Flight

It has been reported in a number of studies like Gao et al. (2018a, 2018b) that exploration is important for solving large-scale and complex optimization problems. In fact, a reduction in the diversity of candidate solutions results in slow convergence speed and entrapment in local minima. Moreover, GSA also has issues with exploration while solving multi-dimensional and complex engineering benchmarks as reported in Rather et al. (2020a). To alleviate the exploration issue of GSA, levy flight distribution has been embedded with the gravitational constant of GSA. Basically, previous studies (Jensi et al., 2016; Li et al., 2016) have shown ample experimental evidence in favor of levy flight for solving diversity issues in HAs.

Generally, levy flight is a random walk having levy distribution based step length (Yang 2010). Moreover, the levy distribution can be defined mathematically by a power-law formula as shown in Equation (8).

\[ L(s) \sim s^{-(1-\beta)}, \quad 0 < \beta \leq 2 \] (8)

where \( s \) and \( \beta \) represents step size and levy index, respectively.

In levy flight, the step size is calculated using the Mantegna algorithm (Yang, 2010) as depicted in Equation (9).

\[ S = \frac{u}{|v|^\frac{1}{\beta}} \] (9)

where \( u \) and \( v \) are normal distributions such that

\[ u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \] (10)

Also,

\[ \sigma_u = \left(\frac{1 + \beta}{\beta^\beta} \sin\left(\frac{\pi \beta}{2}\right)\right) \frac{1}{\beta} \] (11)

\[ \sigma_v = 1 \] (12)
In the proposed LCGSA, levy flight is employed to provide stability between diversification and intensification resulting in overcoming of local minima problem. Besides, the infinite variance of levy distribution provides a strong capability to levy flight for the resolution of falling in local minima and fix improper diversification issues in LCGSA. In fact, a large step size in levy flight results in the enhanced exploration of the search space while a small step size provides high convergence of the solutions towards the global optimum.

5.2. Chaos Theory

Chaos theory, broadly speaking is a branch of mathematics dealing with the study of dynamic systems. The chaotic systems are highly sensitive to the changes in the initial conditions. In other words, small transformation in the inputs create heavy variations in the output of the system. It is obvious that chaotic systems have randomized behavior; however deterministic systems also show chaotic patterns. In the past decade, the researchers have utilized stochasticity, ergodicity, and randomness properties of chaotic maps for resolving premature convergence and local search issues in HAs.

In a related work, the performance of PSO was enhanced by introducing chaotic maps in the velocity equation resulting in enhanced exploitation (Alatas et al., 2009). Likewise, Gandomi et al. (2013) came up with chaos-based accelerated PSO. Similarly, different types of chaotic maps have been utilized in the performance improvement of other HAs such as chaotic GA (Yao et al., 2001), chaotic DE (Zhenyu et al., 2019), and chaotic BBO (Saremi et al., 2014).

Moreover, chaotic maps have been combined with the Imperialist Competitive Algorithm (ICA) to solve real-world problems (Talatahari et al., 2012). Furthermore, the recent additions into the list of enhanced chaotic maps are chaotic bat (Gandomi et al., 2014) and chaotic cuckoo search (Wang et al., 2015) algorithms. These studies confirm the effectiveness and also show the applicability of the chaotic maps in HAs.

In this work, ten chaotic maps have been utilized to enhance the performance of GSA by resolving its premature convergence and exploitation issues. The ten chaotic maps are provided in Table 2. Also, it can be clearly seen that chaotic maps show random behavior as shown in Figure 1.

The initialization of the chaotic maps falls in the range of (0,1) and (-1,1). Besides, $l_{i+1}$ represents the x-axis of the map while i is the chaotic index. As chaotic maps are highly sensitive to the changes in the initial conditions, so a careful selection of initial values is necessary for optimal performance.
## Table 2. Chaotic Maps Combined with GSA

| S. No. | Chaotic Function | Chaotic Map | Limits |
|--------|-----------------|-------------|--------|
| 1.     | Chebyshev (Wang et al., 2015) | $l_{i+1} = \cos(i \cos^{-1}(l_i))$ | (1,-1) |
| 2.     | Circle (Yao et al., 2001) | $l_{i+1} = \mod(l_i + q \cdot \frac{p}{2\pi} \sin(2\pi l_i)), 1)$, p=0.5 and q=0.2 | (0,1) |
| 3.     | Gauss (Jothiprakesh et al., 2013) | $l_{i+1} = \begin{cases} -l_i, & l_i = 0 \\ \frac{1}{\mod(l_i,1)}, & \text{otherwise} \end{cases}$ | (0,1) |
| 4.     | Iterative (Zhenyu et al., 2019) | $l_{i+1} = \sin(\frac{p\pi}{l_i}), p=0.7$ | (-1,1) |
| 5.     | Logistic (Zhenyu et al., 2019) | $l_{i+1} = p \cdot l_i \cdot (1 - l_i), p=4$ | (0,1) |
| 6.     | Piecewise (Saremi et al., 2014) | $l_{i+1} = \begin{cases} \frac{l_i}{r}, & 0 \leq l_i \leq 0 \\ \frac{l_i - r}{0.5 - r}, & 0.5 \leq l_i \leq 0.5 \\ \frac{1 - r - l_i}{0.5 - r}, & 0.5 \leq l_i \leq 1 - r \\ \frac{1 - l_i}{r}, & 1 - r \leq l_i \leq 1 \end{cases}$, r=0.4 | (0,1) |
| 7.     | Sine (Wang et al., 2015) | $l_{i+1} = \frac{p}{4} \sin(\pi l_i), p=4$ | (0,1) |
| 8.     | Singer (Gandomi et al., 2013) | $l_{i+1} = z \cdot (7.86 \cdot l_i^2 - 23.31 \cdot l_i^2 + 28.75 \cdot l_i^3 - 13.302875 \cdot l_i^4)$, z=1.07 | (0,1) |
| 9.     | Sinusoidal (Gandomi et al., 2013) | $l_{i+1} = p \cdot l_i^2 \sin(\pi l_i), p=2.3$ | (0,1) |
| 10.    | Tent (Gandomi et al., 2013) | $l_{i+1} = \begin{cases} \frac{l_i}{0.7}, & l_i \leq 0.7 \\ \frac{10}{3} \cdot (1 - l_i), & l_i \geq 0.7 \end{cases}$ | (0,1) |
Therefore, in this work, we have used the same values for all the ten chaotic maps as specified in Gandomi et al. (2013).

5.3. Mathematical Model of the LCGSA Algorithm

In this section, a modified version of standard GSA has been proposed based on two mathematical tools, that is, levy flight and chaos theory. As stated earlier, the GSA has the drawbacks of skipping true solutions during the optimization process, slow convergence speed, and entrapment in local minima. To resolve the aforementioned issues, two strategies have been employed.

In the first strategy, levy flight distribution has been utilized to solve the diversity issue of standard GSA. In fact, infinite variance and variable step size of levy distribution helps in the resolution of the local optima problem. In other words, it increases the diversity and domain of the search process. The levy flight is mathematically calculated as,

\[
\text{Levy (d, N)} = c' \frac{u}{|\beta|} \sigma_u \frac{u}{\beta} \tag{13}
\]

where d is the dimension of the search space and N is the number of candidate solutions. Moreover, \(c'\) is a multiplicative constant having a value of 0.01, u, and v are normal distributions and \(\beta\) is a levy index with a value of 1.5 (in this work). Moreover, \(\sigma_u\) is given by Equation (11).

In the second strategy, ten different chaotic maps have been employed to overcome slow convergence and the local searching issue of standard GSA. Chaotic maps create huge changes in the output when the initial conditions of the map(s) are modified. This helps searcher agents to move out of the local minima traps. Besides, chaotic normalization (Mirjalili et al., 2017) helps in the proper balance between exploration and exploitation. It is mathematically calculated as shown in Equation (14).

\[
C_{\text{norm}} (t) = \frac{(C_i (t) - a) \ast (d - c)}{(b - a)} + c \tag{14}
\]

In Equation (14), (a, b) is the range of the chaotic map, \(i\) represents chaotic index, and (c, d) are chaotic normalized intervals in which c is having a value of zero while d is calculated using Equation (15).

\[
d = MI - \frac{CI}{MI} (\text{Max-Min}) \tag{15}
\]

Here, MI and CI represent the maximum number of iterations and the current iteration. Besides, adaptive intervals are indicated by Max and Min with a value of 20 and 1e-10.

In standard GSA, the gravitational constant (G) is the main parameter that specifies the intensity of the gravitational field as shown in Equation (2), that is, \(G (t) = G \left( t_0 \right) e^{-\frac{CI}{MI}} \). It is pivotal for providing the proper balance between exploration and exploitation phases. In fact, during the initial iteration phase, the value of G decreases exponentially which results in the exploration of the search space. Moreover, at last iterations, the value of G changes slowly hence, promoting the exploitation of the candidate solutions towards the global optimum. That is why we have chosen G as it is a central controlling parameter of standard GSA which streamlines the exploration and exploitation phases.
In the proposed LCGSA, the levy flight and chaotic sequences have been combined with the gravitational constant of GSA. Hence, Levy-Chaotic gravitational constant \( G^{LC}(t) \) is the addition of Equation(s) (2), (13), and (14).

\[
G^{LC}(t) = \text{Levy (d, N)} + C_{i}^{\text{norm}(t)} + G \left( t_{0} \right) e^{\left[ \pm \frac{CI}{MI} \right]} \tag{16}
\]

Equation (16) shows that \( G^{LC}(t) \) has the interesting properties of levy randomness, chaotic stochasticity, and adaptive learning capability. Broadly speaking, \( G^{LC}(t) \) has all the essential characteristics required for solving entrapment in local minima, intensification, and diversification issues of GSA. The pseudo-code of LCGSA is presented in Algorithm 2. Besides, a flowchart is provided in Figure 2.

Figure 2. Flow Chart of LCGSA Algorithm
Algorithm 2: Levy Flight and Chaos theory Based Gravitational Search Algorithm (LCGSA)

1: Initialize the lower and upper limits of the solution space
2: Evaluate the fitness of each point mass
3: Initialize the LCGSA parameters including the maximum number of iterations \( T \), coefficient \( \alpha \), chaotic index \( \beta \), levy multiplicative constant \( c' \), adaptive intervals: Max and Min
4: Start the iteration number at \( t=0 \)
5: \( \text{while } t < T \text{ do} \)
6: \( \text{for each candidate solution do} \)
7: Calculate Levy \((d, N)\) using Equation (13)
8: Find the chaotic behavior, \( C_i^{\text{norm}}(t) \) of each point mass using Equation (14)
9: Update the Levy-Chaotic gravitational constant, \( G^C(t) \)
10: Calculate the gravitational force between point masses, \( F_i^d(t) \) using Equation (4)
11: Calculate the mass acceleration of the point masses, \( a_i^d(t) \) by using Equation (5)
12: Update the velocity of the point mass solution, \( v_i^d(t+1) \) using Equation (6)
13: Update the position of the point mass solution, \( x_i^d(t+1) \)

Table 3. Unimodal Benchmark Functions

| Function     | Dimension | Search Space |
|--------------|-----------|--------------|
| \( F1(x) = \text{Sphere} \) | 30        | [-100,100]   |
| \( F2(x) = \text{Schwefel 2.22} \) | 30        | [-10,10]     |
| \( F3(x) = \text{Schwefel 1.2} \) | 30        | [-100,100]   |
| \( F4(x) = \text{Schwefel 2.21} \) | 30        | [-100,100]   |
| \( F5(x) = \text{Rosenbrock} \) | 30        | [-30,30]     |
| \( F6(x) = \text{Step} \) | 30        | [-100,100]   |
| \( F7(x) = \text{Quadratic} \) | 30        | [-1.28,1.28] |
using Equation (7)

14: end for
15: t = t + 1
16: end while
17: Return the optimal heavy point mass

Table 4. Multimodal Benchmark Functions

| Function       | Dimension | Search Space   |
|----------------|-----------|----------------|
| F8(x) = Schwefel | 30        | [-500,500]     |
| F9(x) = Rastrigin | 30        | [-5.12,5.12]  |
| F10(x) = Ackley | 30        | [-32,32]       |
| F11(x) = Griewank | 30        | [-600,600]     |
| F12(x) = Penalized | 30        | [-50,50]       |
| F13(x) = Penalize2 | 30        | [-50,50]       |
| F14(x) = Foxholes | 2         | [-65.536,65.536] |
6. ANALYSIS OF SIMULATION RESULTS

The proposed LCGSA is an improved version of standard GSA in which the quality of candidate solutions is increased through diversification and local searching of the search space. Generally, it has been reported that before applying any HA to complex real-world problems; it is recommended to first analyze and benchmark them on some quality test functions.

Therefore, in this section, LCGSA has been applied to 23 well-known classical benchmark functions to test its convergence power, diversity of solutions, and the ability to get away from local minima.

Table 5. Multimodal Benchmark Functions with Fixed Dimension

| Function                   | Dimension | Search Space |
|----------------------------|-----------|--------------|
| F15(x) = Kowalik           | 4         | [-5,5]       |
| F16(x) = Six-hump Camel-back | 2         | [-5,5]       |
| F17(x) = Branin            | 2         | [-5,5]       |
| F18(x) = Goldstein-Price   | 2         | [-2,2]       |
| F19(x) = Hartmen 3         | 3         | [1,3]        |
| F20(x) = Hartmen 6         | 6         | [0,1]        |
| F21(x) = Shekel 5          | 4         | [0,10]       |
| F22(x) = Shekel 7          | 4         | [0,10]       |
| F23(x) = Shekel 10         | 4         | [0,10]       |
6.1. Benchmark Test Functions and Parameter Setting

It is a good practice to test the performance of an HA by applying it to standard benchmark functions. In fact, the benchmark suite consists of functions having variable dimensions and multiple local minima which are ideal for checking the exploration and exploitation capabilities of HAs. Therefore, to check the performance and effectiveness of LCGSA, we have used 23 benchmark functions in which the first 7 are unimodal, the next 7 problems are multi-modal and the last 9 functions are multi-modal with fixed dimensions as shown in Table 3, Table 4, and Table 5, respectively. Moreover, a fixed dimension of 30 is kept for unimodal functions. Broadly speaking, unimodal functions have a single local optimum whereas multi-modal functions have more than one local optima. The benchmark functions are represented by F letter and are numbered as F1, F2, F3, …, F23. Basically, the classical benchmark suite is quite popular in the optimization researcher community as it has been used in a number of studies to test the performance of HAs. The examples include Rashedi et al. (2009), Mirjalili and Lewis (2016), Mirjalili (2017), and so on. In addition, the two-dimensional versions of unimodal and multi-modal functions have been shown in Figure (s) 3, 4, and 5.

The overall experimental analysis from implementation to simulation results has been performed on a computer system having 64 bit OS (Windows 10), 4GB RAM, 2.20 GHz Intel ® core i5 processor, and one TB hard disk. Besides, source code implementation has been carried through MATLAB R2013a. Furthermore, MATLAB codes available online on the GitHub public platform at https://github.com/SajadAHMAD1.

The comparative analysis of experimental results between standard GSA and LCGSA is performed considering the uniform number of searcher agents (50) and the maximum number of iterations (500). Moreover, the HAs were repeated 20 times in order to calculate different statistical measures like average, Standard Deviation (SD), best, worst, and median. The statistical analysis has been performed considering the stochastic and random nature of simulation results. Furthermore, LCGSA 1 to LCGSA 10 are ten Levy-chaotic versions of GSA. In addition, the initialization of different parameters in GSA and LCGSA is shown in Table 6.

Table 6. Parameter Initialization of Standard GSA and LCGSA

| Algorithm | Name of the Parameter | Value of the Parameter |
|-----------|-----------------------|------------------------|
| GSA       | Population Size       | 50                     |
|           | Maximum Number of Iterations | 500                |
|           | Total Number of Runs  | 20                     |
|           | Coefficient (α)       | 20                     |
|           | Initial Value of the Gravitational Constant (G(t₀)) | 100          |
| LCGSA     | Chaotic Index (β)     | 1.5                    |
|           | Multiplicative Constant (c') | 0.01                |
|           | Upper Adaptive Interval (Max) | 20              |
|           | Lower Adaptive Interval (Min) | 1e-10            |
Figure 3. 2D Landscape of Unimodal Benchmark Functions (F₁-F₇)

Figure 4. 2D Landscape of Multi-Modal Benchmark Functions (F₈-F₁₄)

Figure 5. 2D Landscape of Fixed Dimension Multi-Modal Benchmark Functions (F₁₅-F₂₃)
6.2. Comparison of Results between Standard GSA and LCGSA

To validate and test the efficacy of the proposed LCGSA algorithm, it has been applied to the classical benchmark suite of 23 test functions. Moreover, the experimental results of LCGSA will be compared with standard GSA. The simulation results of 20 independent runs are reported to authenticate the results. Besides, various statistical measures including average, worst, best, SD, and median are reported.

6.2.1 Unimodal Test Functions

The benchmark functions (F1-F7) are unimodal which means they have only one local optimum. The unimodal functions are used to test the exploitation and local searching capability of an optimization algorithm. The simulation results of F1-F7 are reported in Table 7 to 13, respectively. It can be clearly seen that LCGSA provides efficient results in the form of average objective function and SD values to

| Algorithm | Best      | Worst     | Average    | SD         | Median     |
|-----------|-----------|-----------|------------|------------|------------|
| GSA       | 1.6011e-17| 6.6336e-17| 3.8664e-17 | 1.2724e-17 | 3.7205e-17 |
| LCGSA1    | 4.3297e-05| 1.25e-08  | 1.0125e-08 | 4.1472e-05 | 1.93e-08   |
| LCGSA2    | 3.2882e-05| 1.6915e-08| 8.3022e-05 | 3.7884e-05 | 7.5934e-05 |
| LCGSA3    | 1.461e-05 | 1.2357e-08| 5.9508e-05 | 3.4397e-05 | 4.6862e-05 |
| LCGSA4    | 6.852e-05 | 8.3722e-08| 1.7405e-08 | 1.6529e-08 | 1.3712e-08 |
| LCGSA5    | 1.2641e-08| 9.8391e-08| 3.0027e-08 | 1.8148e-08 | 2.8984e-08 |
| LCGSA6    | 2.8854e-05| 1.5254e-08| 6.1319e-05 | 2.8947e-05 | 5.5053e-05 |
| LCGSA7    | 3.7724e-05| 1.833e-07 | 8.353e-05  | 3.7699e-05 | 7.2814e-05 |
| LCGSA8    | 2.0244e-05| 1.0537e-08| 5.4595e-05 | 2.798e-05  | 4.1317e-05 |
| LCGSA9    | 1.5295e-05| 1.3938e-08| 4.5315e-05 | 2.5067e-05 | 4.2499e-05 |
| LCGSA10   | 6.1285e-05| 2.9639e-08| 1.7391e-08 | 7.4291e-05 | 1.7035e-08 |

| Algorithm | Best      | Worst     | Average    | SD         | Median     |
|-----------|-----------|-----------|------------|------------|------------|
| GSA       | 2.6472e-08| 6.3052e-06| 3.1526e-07 | 1.4099e-06 | 3.6154e-08 |
| LCGSA1    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA2    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA3    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA4    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA5    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA6    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA7    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA8    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA9    | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
| LCGSA10   | 2.4861e-06| 5.1598e-06| 3.784e-05  | 7.806e-07  | 3.7875e-06 |
four of the seven unimodal test functions. Besides, LCGSA presents competitive results to the other three functions too. Therefore, it can be concluded that LCGSA has outperformed standard GSA as far as convergence power and intensification of the solution space is concerned.

Table 9. Statistical Results of Benchmark Function F3

| Algorithm | Best    | Worst   | Average  | SD      | Median  |
|-----------|---------|---------|----------|---------|---------|
| GSA       | 263.2945| 860.4398| 553.5675 | 170.7562| 531.3584|
| LCGSA1    | 7.5027e-06 | 5.1171  | 1.5422   | 1.4634  | 0.86999 |
| LCGSA2    | 7.5027e-06 | 5.1171  | 1.5422   | 1.4634  | 0.86999 |
| LCGSA3    | 7.5027e-06 | 5.1171  | 1.5422   | 1.4634  | 0.86999 |
| LCGSA4    | 7.5027e-06 | 5.1171  | 1.5422   | 1.4634  | 0.86999 |
| LCGSA5    | 7.5027e-06 | 5.1171  | 1.5422   | 1.4634  | 0.86999 |
| LCGSA6    | 7.5027e-06 | 5.1171  | 1.5422   | 1.4634  | 0.86999 |
| LCGSA7    | 0.21704   | 2.5658  | 0.84421  | 0.61303 | 0.61911 |
| LCGSA8    | 0.21704   | 2.5658  | 0.84421  | 0.61303 | 0.61911 |
| LCGSA9    | 0.21704   | 2.5658  | 0.84421  | 0.61303 | 0.61911 |
| LCGSA10   | 3.8094    | 78.1992 | 18.3088  | 16.3778 | 13.3937 |

Table 10. Statistical Results of Benchmark Function F4

| Algorithm | Best       | Worst      | Average   | SD      | Median  |
|-----------|------------|------------|-----------|---------|---------|
| GSA       | 0.25754    | 7.9535     | 3.4277    | 1.6983  | 3.6703  |
| LCGSA1    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA2    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA3    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA4    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA5    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA6    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA7    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA8    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA9    | 4.676e-16  | 1.1221e-06 | 7.5724e-07| 1.8375e-07| 7.6721e-07|
| LCGSA10   | 6.4536e-07 | 1.2385e-06 | 8.952e-06 | 1.4448e-07| 8.9216e-07|
6.2.2 Multi-modal Test Functions (F8-F23)

The multi-modal problems are those functions in which there is more than one extreme local minima point(s). These problems are used to test the exploration power, global searching capability, and ability to overcome the local minima difficulty of an HA. The simulation results of multimodal functions from F8 to F23 have been presented in Table 14 to 29, respectively. The statistical results imply that LCGSA shows optimal average and SD values to 11 (F8, F9, F11, F12, F13, F15, F17, F19, F21, F22, F23) multimodal functions. However, standard GSA depicts competitive average and SD values to only three (F10, F14, F20) test functions. Besides, both of them show the same performance on F16 and F18. So, it can be inferred from the experimental results that LCGSA has demonstrated efficient quantitative statistical results indicating its global optimization potential, enhanced diversification power, and alleviation from the local minima problem.

### Table 11. Statistical Results of Benchmark Function F5

| Algorithm | Best   | Worst   | Average | SD    | Median |
|-----------|--------|---------|---------|-------|--------|
| GSA       | 25.3344| 35.772  | 27.6865 | 2.08  | 27.3271|
| LCGSA1    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA2    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA3    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA4    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA5    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA6    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA7    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA8    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA9    | 25.5575| 26.9789 | 26.4427 | 0.38831| 26.453 |
| LCGSA10   | 26.581 | 155.114 | 33.5165 | 28.6225 | 27.1498|

### Table 12. Statistical Results of Benchmark Function F6

| Algorithm | Best        | Worst        | Average     | SD        | Median    |
|-----------|-------------|--------------|-------------|-----------|-----------|
| GSA       | 1.9295e-17  | 6.8623e-17   | 3.894e-17   | 1.4463e-17| 3.777e-17 |
| LCGSA1    | 4.6674e-05  | 2.2838e-08   | 1.0815e-08  | 4.4512e-05| 1.0344e-08|
| LCGSA2    | 2.353e-05   | 1.2329e-08   | 6.6547e-05  | 2.6825e-05| 6.2978e-05|
| LCGSA3    | 1.5864e-05  | 1.855e-07    | 5.506e-05   | 4.2458e-05| 4.0269e-05|
| LCGSA4    | 5.2567e-05  | 2.1241e-08   | 1.2466e-08  | 4.808e-05 | 1.2602e-06|
| LCGSA5    | 1.1556e-08  | 9.8071e-08   | 3.4505e-08  | 1.9871e-08| 2.9427e-08|
| LCGSA6    | 2.2696e-05  | 1.1338e-08   | 6.0827e-05  | 2.525e-05 | 5.7602e-05|
| LCGSA7    | 3.413e-05   | 1.8962e-08   | 9.8211e-05  | 4.386e-05 | 9.3959e-05|
| LCGSA8    | 2.4466e-05  | 1.004e-07    | 4.9929e-05  | 1.667e-05 | 4.8592e-05|
| LCGSA9    | 2.6282e-05  | 8.1794e-05   | 5.1028e-05  | 1.8614e-05| 4.5403e-05|
| LCGSA10   | 4.5341e-05  | 2.4746e-08   | 1.2888e-08  | 5.7221e-05| 1.1822e-08|
Theoretically speaking, the dominant performance of LCGSA as compared to standard GSA is due to the "Superposition Effect (SE)" of levy flights and chaotic maps. The SE is basically the ability of levy flight distribution and chaotic maps to take the candidate solutions from infeasible local minima regions to rich feasible optimal solutions. Furthermore, when candidate solutions get trapped in local minima, the variable step size and infinite variance properties of levy flight comes to the rescue. Besides, chaotic maps utilize their stochastic, ergodic patterns and randomized behavior to help candidate solutions to get away from local minima traps and hence, increases the diversification power of the LCGSA.

| Algorithm | Best    | Worst    | Average | SD      | Median  |
|-----------|---------|----------|---------|---------|---------|
| GSA       | 1.7661e-06 | 1.6071e-05 | 4.8999e-06 | 3.7768e-06 | 3.8461e-06 |
| LCGSA1    | 3.0106e-06 | 8.4338e-06 | 5.1864e-06 | 1.3808e-06 | 4.7042e-06 |
| LCGSA2    | 3.1411e-06 | 9.4444e-06 | 5.0383e-06 | 1.4437e-06 | 5.142e-05  |
| LCGSA3    | 9.6553e-07 | 2.9447e-06 | 1.6544e-06 | 5.2305e-07 | 1.5844e-06 |
| LCGSA4    | 3.0816e-06 | 7.1946e-06 | 4.3668e-06 | 1.0932e-06 | 4.0223e-06 |
| LCGSA5    | 1.7627e-06 | 8.1152e-06 | 4.9963e-06 | 1.404e-05  | 4.8246e-06 |
| LCGSA6    | 1.8367e-06 | 6.8152e-06 | 4.6056e-06 | 1.5374e-06 | 4.3952e-06 |
| LCGSA7    | 2.2607e-06 | 7.7719e-06 | 4.7503e-06 | 1.5497e-06 | 4.6268e-06 |
| LCGSA8    | 2.298e-05  | 8.3769e-06 | 5.0295e-06 | 1.569e-05  | 5.0612e-06 |
| LCGSA9    | 2.3069e-06 | 7.9644e-06 | 5.3066e-06 | 1.607e-05  | 5.471e-06  |
| LCGSA10   | 2.6696e-06 | 8.3402e-06 | 5.3426e-06 | 1.7425e-06 | 5.3472e-06 |

Table 14. Statistical Results of Benchmark Function F8

| Algorithm | Best    | Worst    | Average | SD      | Median  |
|-----------|---------|----------|---------|---------|---------|
| GSA       | -3958.02887 | -2041.2984 | -2948.2512 | 420.308 | -2896.9909 |
| LCGSA1    | -3697.6245  | -2085.1277 | -2790.2574 | 425.2239 | -2665.9549 |
| LCGSA2    | -3524.0819  | -1917.8892 | -2709.4617 | 483.2647 | -2726.7727 |
| LCGSA3    | -3810.0014  | -2078.0797 | -2780.4068 | 508.6074 | -2738.1922 |
| LCGSA4    | -3577.978   | -2260.9711 | -2769.7054 | 330.6373 | -2671.6116 |
| LCGSA5    | -3916.656   | -2099.4377 | -2792.8972 | 456.8039 | -2732.4714 |
| LCGSA6    | -3869.9767  | -2071.8347 | -2744.379  | 418.9516 | -2703.1529 |
| LCGSA7    | -3865.1765  | -2128.8874 | -2906.1975 | 529.1391 | -2788.2452 |
| LCGSA8    | -4001.2011  | -2380.302  | -2983.9786 | 428.8163 | -2961.8966 |
| LCGSA9    | -3482.5512  | -2214.1429 | -2783.7013 | 374.4724 | -2660.2603 |
| LCGSA10   | -3338.8703  | -2392.5634 | -2715.9929 | 229.9425 | -2680.0465 |
6.3. Statistical Analysis between Standard GSA and LCGSA

The minimum value of mean and standard deviation does not imply that an algorithm is efficient than others (Derrac et al., 2011). However, statistical tests should be performed on the simulation results to find the optimal competitive algorithm. Therefore, a pairwise non-parametric signed Wilcoxon rank-sum test has been performed at a 5% significance level to statistically validate the simulation results between GSA and LCGSA. The reason behind selecting a Wilcoxon rank-sum test is that it uses median as a statistical measure which is better than SD. Moreover, in the Wilcoxon rank-sum test the distribution of the dataset is not considered.

The null hypothesis consists of the best performing algorithm having p-values greater than 0.05 whereas the alternate hypothesis includes algorithm with p-values less than 0.05. Besides, a p-value of 1 shows the statistical equivalency of competing algorithms. Moreover, the standard GSA has

| Algorithm | Best      | Worst     | Average  | SD        | Median  |
|-----------|-----------|-----------|----------|-----------|---------|
| GSA       | 8.9546    | 32.8336   | 19.4017  | 5.6144    | 18.4067 |
| LCGSA1    | 10.9584   | 26.8795   | 20.3671  | 4.4056    | 19.918  |
| LCGSA2    | 8.9724    | 23.8911   | 17.7736  | 4.2354    | 18.9123 |
| LCGSA3    | 10.9497   | 28.8617   | 18.6197  | 6.3499    | 16.9253 |
| LCGSA4    | 10.9686   | 27.8746   | 18.9777  | 3.9732    | 18.9234 |
| LCGSA5    | 10.0068   | 22.9571   | 17.388   | 4.2723    | 16.9523 |
| LCGSA6    | 9.9794    | 22.8982   | 17.3757  | 4.0722    | 17.919  |
| LCGSA7    | 9.9666    | 25.883    | 17.036   | 3.5772    | 15.9493 |
| LCGSA8    | 10.9484   | 39.8067   | 19.5103  | 7.1034    | 18.4171 |
| LCGSA9    | 12.9431   | 29.8609   | 21.698   | 4.3148    | 21.8969 |
| LCGSA10   | 11.9761   | 26.8901   | 18.5869  | 4.5494    | 17.9221 |

Table 15. Statistical Results of Benchmark Function F9

| Algorithm | Best      | Worst     | Average  | SD        | Median  |
|-----------|-----------|-----------|----------|-----------|---------|
| GSA       | 3.9927e-09| 5.7138e-09| 4.657e-09| 5.6026e-10| 4.5181e-09|
| LCGSA1    | 5.0646e-07| 1.186e-05 | 7.6598e-07| 1.6819e-07| 7.4542e-07|
| LCGSA2    | 3.4533e-07| 1.0085e-06| 6.0136e-07| 1.7636e-07| 5.8771e-07|
| LCGSA3    | 3.7425e-07| 1.2299e-06| 6.3102e-07| 1.9388e-07| 6.1049e-07|
| LCGSA4    | 5.2679e-07| 1.3502e-06| 9.5959e-07| 2.3675e-07| 9.7996e-07|
| LCGSA5    | 9.619e-06  | 2.5833e-06 | 1.4927e-06 | 3.8722e-07 | 1.4603e-06 |
| LCGSA6    | 4.1117e-07| 8.9283e-07 | 6.0972e-07 | 1.1117e-07 | 6.113e-06   |
| LCGSA7    | 4.976e-06 | 1.2953e-06 | 7.6685e-07 | 1.8989e-07 | 7.662e-06   |
| LCGSA8    | 3.3769e-07| 8.0165e-07 | 5.2756e-07 | 1.0844e-07 | 5.458e-06   |
| LCGSA9    | 2.8501e-07| 6.9042e-07 | 4.9899e-07 | 1.1041e-07 | 4.8968e-07  |
| LCGSA10   | 6.5164e-07| 1.4273e-06 | 1.0268e-06 | 2.111e-06   | 1.0503e-06  |

Table 16. Statistical Results of Benchmark Function F10
been compared with the best performing LCGSA version which is having less mean and SD values. Table 30 shows the Wilcoxon rank-sum test p-values of standard GSA and LCGSA. Furthermore, ‘H’ represents LCGSA is statistically significant than standard GSA, ‘L’ indicates standard GSA is efficient than LCGSA whereas ‘E’ specifies that both LCGSA and GSA are statistically equivalent. The p-values indicate that LCGSA provides efficient statistical results on 16 benchmark problems whereas standard GSA gives better statistical results on only five test functions. Besides, both LCGSA and GSA show the same statistical performance on two benchmark functions. In short, the results of the Wilcoxon statistical test specify that LCGSA is an efficient optimizer than GSA.

### Table 17. Statistical Results of Benchmark Function F11

| Algorithm | Best       | Worst      | Average  | SD       | Median  |
|-----------|------------|------------|----------|----------|---------|
| GSA       | 9.0598     | 24.9689    | 17.6925  | 3.9121   | 17.7883 |
| LCGSA1    | 7.398e-06  | 3.6834     | 1.9056   | 0.98993  | 1.6363  |
| LCGSA2    | 1.7146     | 7.4542     | 3.4618   | 1.5093   | 3.0751  |
| LCGSA3    | 6.4607     | 19.8159    | 12.9456  | 3.6522   | 13.0077 |
| LCGSA4    | 1.1544     | 4.8144     | 2.4633   | 0.87363  | 2.262   |
| LCGSA5    | 1.0557     | 5.4978     | 2.8915   | 1.2062   | 2.7189  |
| LCGSA6    | 1.1856     | 5.3377     | 2.7535   | 1.272    | 2.4461  |
| LCGSA7    | 1.3894     | 6.2781     | 3.3875   | 1.4549   | 3.1961  |
| LCGSA8    | 8.1006e-06 | 2.4652     | 1.2366   | 0.52774  | 1.3325  |
| LCGSA9    | 8.9253e-07 | 1.6721     | 0.59454  | 0.56746  | 0.39047 |
| LCGSA10   | 1.1761     | 5.4018     | 2.8284   | 1.3979   | 2.4396  |

### Table 18. Statistical Results of Benchmark Function F12

| Algorithm | Best       | Worst      | Average  | SD       | Median  |
|-----------|------------|------------|----------|----------|---------|
| GSA       | 1.6437e-19 | 1.8775     | 0.52979  | 0.50709  | 0.31276 |
| LCGSA1    | 1.9906e-07 | 0.10367    | 2.5082e-06 | 4.2738e-06 | 8.3581e-07 |
| LCGSA2    | 2.353e-07  | 0.207032   | 4.6649e-06 | 7.1146e-06 | 5.1376e-07 |
| LCGSA3    | 1.0443e-07 | 1.2478     | 0.21419  | 0.36707  | 0.10367 |
| LCGSA4    | 2.7034e-07 | 0.6227     | 4.1503e-06 | 0.14045  | 8.3327e-07 |
| LCGSA5    | 8.8514e-07 | 0.20732    | 2.8032e-06 | 5.6799e-06 | 2.0626e-06 |
| LCGSA6    | 2.072e-07  | 0.31096    | 3.1099e-06 | 7.5949e-06 | 3.87e-07  |
| LCGSA7    | 2.8914e-07 | 0.41467    | 5.4762e-06 | 0.1083   | 5.8938e-07 |
| LCGSA8    | 1.5892e-07 | 0.20732    | 4.665e-05  | 6.2697e-06 | 5.1361e-07 |
| LCGSA9    | 1.5009e-07 | 0.58327    | 7.063e-05  | 0.13531  | 4.4983e-07 |
| LCGSA10   | 3.6324e-07 | 0.31096    | 4.1176e-06 | 8.3821e-06 | 1.1662e-06 |
Table 19. Statistical Results of Benchmark Function F13

| Algorithm | Best         | Worst        | Average | SD     | Median   |
|-----------|--------------|--------------|---------|--------|----------|
| GSA       | 9.9034e-08   | 5.3339       | 1.86    | 1.7349 | 1.508    |
| LCGSA1    | 3.6454e-06   | 1.0997e-06   | 1.1078e-07 | 3.3818e-07 | 7.558e-06 |
| LCGSA2    | 3.1403e-06   | 0.011        | 1.6555e-07 | 4.025e-06 | 7.5424e-06 |
| LCGSA3    | 2.4039e-06   | 1.0995e-06   | 5.55e-06  | 2.4573e-07 | 4.4425e-06 |
| LCGSA4    | 4.2929e-06   | 8.6241e-07   | 4.4048e-08 | 1.926e-07 | 9.4367e-06 |
| LCGSA5    | 1.37e-05     | 1.1013e-06   | 5.7889e-08 | 2.456e-06 | 3.0458e05 |
| LCGSA6    | 3.2818e-06   | 0.011        | 1.1055e-07 | 3.3831e-07 | 6.3947e-06 |
| LCGSA7    | 4.8464e-06   | 1.0996e-06   | 5.5775e-08 | 2.4569e-07 | 7.5706e-06 |
| LCGSA8    | 1.6151e-06   | 1.948e-05    | 5.3125e-06 | 2.2149e-06 | 5.07212e-06 |
| LCGSA9    | 1.8295e-06   | 1.0993e-06   | 1.5866e-07 | 3.8727e-07 | 4.1487e-06 |
| LCGSA10   | 6.6605e-06   | 1.1008e-06   | 3.1419e-07 | 4.9522e-07 | 1.6953e-05 |

Table 20. Statistical Results of Benchmark Function F14

| Algorithm | Best         | Worst        | Average | SD     | Median   |
|-----------|--------------|--------------|---------|--------|----------|
| GSA       | 0.998        | 7.7202       | 2.993   | 2.0484 | 2.7371   |
| LCGSA1    | 0.99812      | 12.0818      | 5.0636  | 3.681  | 3.8412   |
| LCGSA2    | 0.99804      | 9.3022       | 3.2332  | 2.6188 | 2.4423   |
| LCGSA3    | 0.998        | 10.7958      | 4.6007  | 3.0356 | 3.9456   |
| LCGSA4    | 0.998        | 10.8303      | 5.7216  | 2.9037 | 6.0064   |
| LCGSA5    | 0.998        | 7.9312       | 3.9054  | 2.2251 | 3.0047   |
| LCGSA6    | 0.998        | 13.6187      | 4.5735  | 3.5481 | 3.1748   |
| LCGSA7    | 1.0003       | 7.9723       | 3.4896  | 2.2142 | 2.9737   |
| LCGSA8    | 0.99801      | 9.2094       | 4.7157  | 2.7066 | 4.064    |
| LCGSA9    | 0.998        | 11.9321      | 3.5853  | 2.9718 | 2.989    |
| LCGSA10   | 1.005        | 9.8418       | 4.9648  | 2.3703 | 4.6903   |
Table 21. Statistical Results of Benchmark Function F15

| Algorithm   | Best         | Worst        | Average   | SD        | Median     |
|-------------|--------------|--------------|-----------|-----------|------------|
| GSA         | 2.4391e-07   | 1.824e-05    | 7.1053e-07| 3.9007e-07| 7.225e-06  |
| LCGSA1      | 7.2371e-08   | 1.0934e-06   | 2.224e-06 | 2.7867e-07| 7.8266e-08 |
| LCGSA2      | 7.2037e-08   | 1.2189e-06   | 3.7559e-07| 3.4965e-07| 2.9446e-07 |
| LCGSA3      | 7.9475e-08   | 8.8143e-07   | 2.5414e-07| 2.357e-06  | 1.2934e-07 |
| LCGSA4      | 7.0928e-08   | 1.6007e-06   | 2.326e-06 | 3.9103e-07| 7.7558e-08 |
| LCGSA5      | 7.6063e-08   | 1.0511e-06   | 3.2517e-07| 3.2512e-07| 1.2235e-07 |
| LCGSA6      | 7.6175e-08   | 9.4084e-07   | 2.4234e-07| 2.6322e-07| 7.8938e-08 |
| LCGSA7      | 7.4384e-08   | 8.3905e-07   | 2.794e-06 | 2.4335e-07| 1.8735e-07 |
| LCGSA8      | 7.4696e-08   | 8.1039e-07   | 2.034e-07 | 2.6126e-07| 7.8266e-08 |
| LCGSA9      | 7.5272e-08   | 1.4427e-06   | 3.5328e-07| 4.4871e-07| 7.8266e-08 |
| LCGSA10     | 6.9791e-08   | 8.3884e-07   | 2.6079e-07| 2.6751e-07| 8.2635e-08 |

Table 22. Statistical Results of Benchmark Function F16

| Algorithm   | Best         | Worst        | Average   | SD        | Median     |
|-------------|--------------|--------------|-----------|-----------|------------|
| GSA         | -1.0316      | -1.0316      | -1.0316   | 1.0188e-06| -1.0316    |
| LCGSA1      | -1.0316      | -1.0316      | -1.0316   | 9.9731e-08| -1.0316    |
| LCGSA2      | -1.0316      | -1.0316      | -1.0316   | 2.7742e-08| -1.0316    |
| LCGSA3      | -1.0316      | -1.0316      | -1.0316   | 1.9744e-08| -1.0316    |
| LCGSA4      | -1.0316      | -1.0316      | -1.0316   | 5.5966e-08| -1.0316    |
| LCGSA5      | -1.0316      | -1.0316      | -1.0316   | 2.5436e-07| -1.0316    |
| LCGSA6      | -1.0316      | -1.0316      | -1.0316   | 3.3425e-08| -1.0316    |
| LCGSA7      | -1.0316      | -1.0316      | -1.0316   | 5.1945e-08| -1.0316    |
| LCGSA8      | -1.0316      | -1.0316      | -1.0316   | 1.777e-08  | -1.0316    |
| LCGSA9      | -1.0316      | -1.0316      | -1.0316   | 1.8489e-08| -1.0316    |
| LCGSA10     | -1.0316      | -1.0316      | -1.0316   | 2.5086e-08| -1.0316    |
Table 23. Statistical Results of Benchmark Function F17

| Algorithm  | Best   | Worst  | Average | SD      | Median |
|------------|--------|--------|---------|---------|--------|
| GSA        | 0.39789| 0.39789| 0.39789 | 0       | 0.39789|
| LCGSA1     | 0.39789| 0.39789| 0.39789 | 2.0635e-08| 0.39789|
| LCGSA2     | 0.39789| 0.39789| 0.39789 | 1.5593e-08| 0.39789|
| LCGSA3     | 0.39789| 0.39789| 0.39789 | 5.1054e-09| 0.39789|
| LCGSA4     | 0.39789| 0.39789| 0.39789 | 1.2216e-08| 0.39789|
| LCGSA5     | 0.39789| 0.39789| 0.39789 | 7.0273e-08| 0.39789|
| LCGSA6     | 0.39789| 0.39789| 0.39789 | 2.1947e-08| 0.39789|
| LCGSA7     | 0.39789| 0.39789| 0.39789 | 1.49e-08  | 0.39789 |
| LCGSA8     | 0.39789| 0.39789| 0.39789 | 1.4032e-08| 0.39789|
| LCGSA9     | 0.39789| 0.39789| 0.39789 | 1.7602e-08| 0.39789|
| LCGSA10    | 0.39789| 0.39789| 0.39789 | 1.5212e-08| 0.39789|

Table 24. Statistical Results of Benchmark Function F18

| Algorithm  | Best | Worst | Average | SD        | Median |
|------------|------|-------|---------|-----------|--------|
| GSA        | 3    | 3     | 3       | 2.312e-15 | 3      |
| LCGSA1     | 3    | 3     | 3       | 1.7882e-06| 3      |
| LCGSA2     | 3    | 3     | 3       | 1.745e-06 | 3      |
| LCGSA3     | 3    | 3     | 3       | 4.9674e-07| 3      |
| LCGSA4     | 3    | 3     | 3       | 9.7783e-07| 3      |
| LCGSA5     | 3    | 3     | 3       | 7.1589e-06| 3      |
| LCGSA6     | 3    | 3     | 3       | 1.0983e-06| 3      |
| LCGSA7     | 3    | 3     | 3       | 2.9504e-06| 3      |
| LCGSA8     | 3    | 3     | 3       | 1.5169e-06| 3      |
| LCGSA9     | 3    | 3     | 3       | 1.6032e-06| 3      |
| LCGSA10    | 3    | 3     | 3       | 1.7649e-06| 3      |
Table 25. Statistical Results of Benchmark Function F19

| Algorithm | Best   | Worst  | Average | SD    | Median |
|-----------|--------|--------|---------|-------|--------|
| GSA       | -3.8628| -2.9716| -3.6475 | 0.27325| -3.7677|
| LCGSA1    | -3.8628| -2.9964| -3.6204 | 0.23999| -3.6504|
| LCGSA2    | -3.8628| -2.8764| -3.6117 | 0.2891 | -3.691 |
| LCGSA3    | -3.8628| -2.8588| -3.6272 | 0.32859| -3.8523|
| LCGSA4    | -3.8628| -2.9524| -3.603  | 0.29353| -3.725 |
| LCGSA5    | -3.8628| -2.7764| -3.5564 | 0.34358| -3.6855|
| LCGSA6    | -3.8628| -2.9068| -3.497  | 0.32455| -3.5358|
| LCGSA7    | -3.8628| -2.568 | -3.4815 | 0.36243| -3.449 |
| LCGSA8    | -3.8628| -2.7975| -3.676  | 0.28686| -3.8373|
| LCGSA9    | -3.8628| -2.7749| -3.5264 | 0.32275| -3.6584|
| LCGSA10   | -3.8628| -3.0685| -3.5437 | 0.24827| -3.603 |

Table 26. Statistical Results of Benchmark Function F20

| Algorithm | Best   | Worst  | Average | SD    | Median |
|-----------|--------|--------|---------|-------|--------|
| GSA       | -3.322 | -1.2547| -2.2248 | 0.71465| -2.031 |
| LCGSA1    | -2.5591| 0.75846| -1.8229 | 0.48091| -1.718 |
| LCGSA2    | -3.322 | -1.2541| -2.0357 | 0.52367| -2.0075|
| LCGSA3    | -3.322 | -1.2728| -2.0937 | 0.68105| -1.7061|
| LCGSA4    | -2.6058| -1.1538| -1.796  | 0.42291| -1.7909|
| LCGSA5    | -2.9967| -1.2328| -2.0053 | 0.48983| -2.0701|
| LCGSA6    | -2.804 | -1.2175| -1.9112 | 0.47638| -1.8612|
| LCGSA7    | -2.6904| -1.1868| -1.6927 | 0.40647| -1.6793|
| LCGSA8    | -2.8858| -0.91831| -1.9371 | 0.57635| -1.9789|
| LCGSA9    | -3.0853| -1.1389| -2.1682 | 0.53021| -2.1018|
| LCGSA10   | -2.7859| 0.91422| -1.768  | 0.58557| -1.5932|
**Table 27. Statistical Results of Benchmark Function F21**

| Algorithm | Best    | Worst   | Average | SD     | Median |
|-----------|---------|---------|---------|--------|--------|
| GSA       | -10.1532 | -2.6829 | -5.1016 | 1.3534 | -5.0552 |
| LCGSA1    | -10.1532 | -3.1399 | -6.2457 | 2.4352 | -5.0552 |
| LCGSA2    | -5.0552  | -5.0552 | -5.0552 | 1.7143E-06 | -5.0552 |
| LCGSA3    | -10.1532 | -3.0192 | -5.2083 | 1.2495 | -5.0552 |
| LCGSA4    | -10.1532 | -2.6829 | -5.3655 | 1.3798 | -5.0552 |
| LCGSA5    | -5.6398  | -4.4604 | -5.0547 | 0.19132 | -5.0552 |
| LCGSA6    | -5.0552  | -5.0552 | -5.0552 | 1.6729e-06 | -5.0552 |
| LCGSA7    | -6.432   | -4.8139 | -5.1068 | 0.31635 | -5.0552 |
| LCGSA8    | -5.0552  | -4.4674 | -5.0258 | 0.13144 | -5.0552 |
| LCGSA9    | -5.0552  | -5.0552 | -5.0052 | -5.0552 | -5.0552 |
| LCGSA10   | -10.1532 | -2.6829 | -5.0761 | 1.3942 | -5.0552 |

**Table 28. Statistical Results of Benchmark Function F22**

| Algorithm | Best    | Worst   | Average | SD     | Median |
|-----------|---------|---------|---------|--------|--------|
| GSA       | -10.4029 | -5.0877 | -8.0111 | 2.713  | -10.4029 |
| LCGSA1    | -10.4029 | -10.4029 | -10.4029 | 8.0615e-06 | -10.4029 |
| LCGSA2    | -10.4029 | -5.0877 | -6.948 | 2.6011 | -5.0877 |
| LCGSA3    | -10.4029 | -2.7659 | -7.895 | 2.8887 | -10.4029 |
| LCGSA4    | -10.4029 | -5.0877 | -8.2768 | 2.6716 | -10.4029 |
| LCGSA5    | -10.4029 | -6.8917 | -10.2273 | 0.78511 | -10.4029 |
| LCGSA6    | -10.4029 | -5.0877 | -8.7487 | 2.4731 | -10.4029 |
| LCGSA7    | -10.4029 | -5.0877 | -8.4307 | 2.5647 | -10.4029 |
| LCGSA8    | -10.4029 | -5.0877 | -8.5426 | 2.6011 | -10.4029 |
| LCGSA9    | -10.4029 | -5.0877 | -7.2138 | 2.6716 | -5.0877 |
| LCGSA10   | -10.4029 | -5.0877 | -7.2138 | 2.6716 | -5.0877 |
### Table 29. Statistical Results of Benchmark Function F23

| Algorithm  | Best     | Worst    | Average | SD       | Median  |
|------------|----------|----------|---------|----------|---------|
| GSA        | -10.5364 | -5.1285  | -9.1844 | 2.4025   | -10.5364|
| LCGSA1     | -10.5364 | -7.4563  | -10.3824| 0.68872  | -10.5364|
| LCGSA2     | -10.5364 | -10.5364 | -10.5364| 6.0152e-06| -10.5364|
| LCGSA3     | -10.5364 | -10.5364 | -10.5364| 3.7914e-06| -10.5364|
| LCGSA4     | -10.5364 | -5.1285  | -10.1712| 1.2601   | -10.5364|
| LCGSA5     | -10.5364 | -6.4041  | -10.5364| 0.92399  | -10.5364|
| LCGSA6     | -10.5364 | -10.5364 | -10.5364| 5.2928e-06| -10.5364|
| LCGSA7     | -10.5364 | -5.1285  | -9.9686 | 1.4692   | -10.5364|
| LCGSA8     | -10.5364 | -10.5364 | -10.5364| 1.0689e-05| -10.5364|
| LCGSA9     | -10.5364 | -10.5364 | -10.5364| 1.1257e-05| -10.5364|
| LCGSA10    | -10.5364 | -10.5364 | -10.5364| 8.6233e-06| -10.5364|

### Table 30. Statistical Analysis of Classical Benchmark Suite

| Benchmark Function | P-value     | Decision |
|--------------------|-------------|----------|
| F1                 | 8.85745e-05 | H        |
| F2                 | 1.03346e-04 | H        |
| F3                 | 8.85745e-05 | H        |
| F4                 | 8.85745e-05 | H        |
| F5                 | 0.00359     | H        |
| F6                 | 8.85745e-05 | H        |
| F7                 | 2.19079e-04 | H        |
| F8                 | 0.68132     | L        |
| F9                 | 0.145399    | L        |
| F10                | 8.85745e-05 | H        |
| F11                | 8.85745e-05 | H        |
| F12                | 2.93161e-04 | H        |
| F13                | 8.85745e-05 | H        |
| F14                | 0.97921     | L        |
| F15                | 3.90231e-04 | H        |
| F16                | 1           | E        |
| F17                | 8.85745e-05 | H        |
| F18                | 1           | E        |
| F19                | 0.190821    | L        |
| F20                | 0.77        | L        |
| F21                | 0.01        | H        |
| F22                | 0.01        | H        |
| F23                | 0           | H        |
6.4 Comparison of Complexity between Standard GSA and LCGSA

The optimization potential of HA is inversely proportional to its computational complexity. In other words, when HA solves a real-world complex problem in less time, it depicts its time efficiency and practical applicability. So, it is essential to determine the time complexity of HA as far as its problem-solving potential is concerned. Therefore, the time complexity of LCGSA has been calculated using big-O notation by considering the pseudo-code as depicted in Algorithm 2. It is important to mention that the initialization parameters like the dimension of the problem (d), population size (n), and stopping criterion, that is, the maximum number of iterations (t) have a significant contribution to the computational complexity of LCGSA. So, the worst-case time complexity of LCGSA is as follows:

| Benchmark Function | Execution Time (in seconds) |
|--------------------|-----------------------------|
|                    | GSA            | LCGSA          |
| F1                 | 113.8878       | 114.3707       |
| F2                 | 96.9693        | 106.0548       |
| F3                 | 141.9151       | 131.9804       |
| F4                 | 109.9906       | 115.0801       |
| F5                 | 104.9648       | 108.9495       |
| F6                 | 112.4235       | 124.2084       |
| F7                 | 104.172        | 108.1329       |
| F8                 | 209.6629       | 118.0783       |
| F9                 | 96.3732        | 102.997        |
| F10                | 118.7088       | 121.712        |
| F11                | 124.1387       | 155.3285       |
| F12                | 174.8475       | 222.5923       |
| F13                | 156.711        | 168.3772       |
| F14                | 122.941        | 131.3308       |
| F15                | 80.6426        | 88.9527        |
| F16                | 111.7934       | 117.5812       |
| F17                | 129.8535       | 215.7595       |
| F18                | 75.2148        | 90.6687        |
| F19                | 82.4062        | 90.6687        |
| F20                | 78.6018        | 69.7136        |
| F21                | 76.4252        | 93.8004        |
| F22                | 90.2827        | 84.174         |
| F23                | 82.1011        | 82.5164        |
O(LCGSA) = O( Levy-Chaotic gravitational constant update) + O(position update phase) + O(velocity update phase) + O(selection of optimal heavy point mass)
  \( = O(t (nd + n)) \)

Where, \( t = \) maximum number of iterations
\( n = \) population size
\( d = \) dimension of the benchmark function

Similarly, \( O(GSA) = O(nd + tn) \)

Therefore, the computational complexity of both LCGSA and GSA is the same. Moreover, the CPU execution time in seconds has been taken for GSA and LCGSA while solving benchmark functions as reported in Table 31. It has been calculated after 20 independent runs by GSA and LCGSA. Furthermore, it indicates that LCGSA takes a little more time as compared to standard GSA. Clearly, LCGSA would have to take more time because it needs the additional number of function evaluations to go through two phases of levy distribution and chaotic maps for optimization of candidate solutions.

6.5. Performance Index (PI) Analysis

The PI has been calculated to compare the simulation results of standard GSA and proposed LCGSA. The concept of PI was given by Deep et al. (2007). Basically, PI is the calculation of the relative performance of an algorithm by giving weights to success rate \( (\alpha_{TP}) \) and execution time \( (\beta_{TP}) \). The PI is mathematically calculated by using Equation (17).

\[
PI = \frac{1}{N_{TP}} \sum_{TP=1}^{N_{TP}} \left( W_{1} \alpha_{TP} + W_{2} \beta_{TP} \right)
\]

Where \( N_{TP} \) is the total number of test problems under consideration. Also, the success rate and execution time are given by Equations (18) and (19).
\[ \alpha_{TP} = \frac{SR_{TP}}{TR_{TP}} \]  \hspace{1cm} (18)

\[ \beta_{TP} = \frac{MT_{TP}}{AT_{TP}} \]  \hspace{1cm} (19)

In Equation (18), \( SR_{TP} \) is the number of successful runs for test problem TP and \( TR_{TP} \) is the total number of runs. Moreover, \( MT_{TP} \) is the minimum execution time taken by all algorithms while solving test function TP and \( AT_{TP} \) is the average time taken by an algorithm for solving TP. Besides, non-negative weights are represented by \( W_1 \) and \( W_2 \) which are related to \( \alpha_{TP} \) and \( \beta_{TP} \) such that \( W_1 + W_2 = 1 \) (\( 0 \leq W_1, W_2 \leq 1 \)). In addition, we have calculated PI considering three different cases of weights.

Figure 7. PI of GSA and LCGSA for Case (II)

Figure 8. PI of GSA and LCGSA for Case (III)
Table 32. Comparative Analysis of LCGSA with other Heuristic Algorithms

| Test Function | GSA  | PSO  | BBO  | GA   | DE   | ACO   | SSA   | SCA   | GWO   | LCGSA |
|---------------|------|------|------|------|------|-------|-------|-------|-------|-------|
| F1            | 3.86e-17 | 1.12e-08 | 1.7506 | 3.6715 | 5.78e-08 | 9455  | 2.06e-08 | 5.3378 | 2.64e-33 | 1.01e-08 |
| F2            | 3.15e-17 | 1.45e-06 | 0.37571 | 1.04037 | 2.66e-07 | 2.65e+32 | 0.76457 | 6.77e-07 | 7.5e-08 | 3.7e-05 |
| F3            | 553.567 | 579.4562 | 265.7637 | 2979.767 | 30172.39 | 1624888 | 548.45 | 7696.855 | 3.85e-08 | 1.5422 |
| F4            | 3.4277 | 3.4805 | 1.0696 | 6.4467 | 12.8927 | 30 | 7.2387 | 33.6344 | 2.65e-08 | 7.57e-07 |
| F5            | 27.6865 | 65.7746 | 211.4699 | 590.435 | 144.5587 | 4138903 | 193.6834 | 13445.44 | 26.8153 | 26.4427 |
| F6            | 3.89e-17 | 1.01e-07 | 1.6228 | 4.417 | 5.82e-08 | 9227.3 | 2.12e-08 | 9.5207 | 0.41797 | 1.08e-08 |
| F7            | 4.89e-06 | 2.06e-06 | 6.69e-07 | 2.61e-06 | 5.03e-06 | 4253951 | 9.74e-06 | 8.22e-06 | 1.11e-07 | 1.65e-06 |
| F8            | -2948.25 | -7806.73 | -8829.20 | -10388.6 | -9488.56 | 314.0043 | -7497.41 | -3808.32 | -6505.25 | -2983.97 |
| F9            | 19.4017 | 101.894 | 36.0419 | 30.1574 | 86.1647 | 9455 | 44.4746 | 35.6045 | 1.9015 | 17.0361 |
| F10           | 4.65e-09 | 4.69e-07 | 0.42183 | 1.876 | 6.64e-07 | 19.4258 | 2.093 | 14.0428 | 4.38e-14 | 5.27e-07 |
| F11           | 17.6925 | 2.15e-06 | 0.91379 | 0.78333 | 9.11e-07 | 3.3632 | 1.46e-06 | 0.83928 | 1.78e-07 | 0.59454 |
| F12           | 0.52979 | 3.11e-06 | 3.65e-06 | 5.63e-06 | 6.94e-05 | 7226686 | 4.9926 | 5.8184 | 2.23e-06 | 2.50e-06 |
| F13           | 1.86 | 4.46e-07 | 7.12e-06 | 0.82564 | 3.73e-08 | 2153653 | 3.1435 | 19242.13 | 0.43805 | 4.40e-08 |
| F14           | 2.993 | 0.998 | 4.0873 | 0.998 | 0.998 | 67.4721 | 1.0972 | 1.5936 | 2.7682 | 3.2332 |
| F15           | 7.10e-07 | 1.89e-07 | 1.90e-07 | 7.25e-07 | 7.89e-08 | 0.40836 | 7.89e-08 | 1.05e-07 | 2.46e-07 | 2.03e-07 |
| F16           | -1.0316 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | 5.7333 | -1.0316 | -1.0316 | -1.0316 | -1.0316 |
| F17           | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39789 | 0.39789 |
| F18           | 3 | 3 | 4.35 | 3 | 3 | 2275 | 3 | 3 | 3 | 3 |
| F19           | -3.6475 | -3.8628 | -3.8628 | -3.8628 | -3.8628 | -1.22e-09 | -3.8628 | -3.8543 | -3.862 | -3.6475 |
| F20           | -2.2248 | -3.2685 | -3.2863 | -3.2923 | -3.3218 | -8.58e-07 | -3.2211 | -3.0164 | -3.2518 | -2.1682 |
| F21           | -5.1016 | -7.7571 | -5.7758 | -7.1467 | -9.8927 | -0.20055 | -8.8986 | -3.3441 | -10.1518 | -5.3655 |
| F22           | -8.0111 | -9.4909 | -7.8462 | -8.4189 | -10.4028 | -0.30383 | -8.8433 | -4.069 | -10.4016 | -8.2768 |
| F23           | -9.1844 | -7.0779 | -6.8484 | -8.1665 | -10.5364 | -0.38314 | -9.7442 | -4.4617 | -10.5347 | -10.5364 |

Figure 9a. Convergence Curves for Unimodal Test Function F1
Figure 9b. Convergence Curves for Unimodal Test Function F2

Figure 9c. Convergence Curves for Unimodal Test Function F3

Figure 9d. Convergence Curves for Unimodal Test Function F4
Figure 9e. Convergence Curves for Unimodal Test Function F5

Figure 9f. Convergence Curves for Unimodal Test Function F6

Figure 9g. Convergence Curves for Unimodal Test Function F7
Figure 9h. Convergence Curves of LCGSA and other HAs for Unimodal Test Function F5

![Convergence Curves of LCGSA and other HAs for Unimodal Test Function F5](image)

Figure 10a. Convergence Curves for Multimodal Test Function F8

![Convergence Curves for Multimodal Test Function F8](image)

Figure 10b. Convergence Curves for Multimodal Test Function F9

![Convergence Curves for Multimodal Test Function F9](image)
Figure 10c. Convergence Curves for Multimodal Test Function F10

Figure 10d. Convergence Curves for Multimodal Test Function F11

Figure 10e. Convergence Curves for Multimodal Test Function F12
Figure 10f. Convergence Curves for Multimodal Test Function F13

Figure 10g. Convergence Curves for Multimodal Test Function F14

Figure 10h. Convergence Curves of LCGSA and other HAs for Multimodal Test Function F12
Figure 11a. Convergence Curves for Multimodal Test Function F15 with Fixed Dimension

Figure 11b. Convergence Curves for Multimodal Test Function F17 with Fixed Dimension

Figure 11c. Convergence Curves for Multimodal Test Function F18 with Fixed Dimension
Figure 11d. Convergence Curves for Multimodal Test Function F19 with Fixed Dimension

Figure 11e. Convergence Curves for Multimodal Test Function F20 with Fixed Dimension

Figure 11f. Convergence Curves for Multimodal Test Function F21 with Fixed Dimension
Figure 11g. Convergence Curves for Multimodal Test Function F22 with Fixed Dimension

![Convergence Curves for Multimodal Test Function F22 with Fixed Dimension](image)

Figure 11h. Convergence Curves for Multimodal Test Function F23 with Fixed Dimension

![Convergence Curves for Multimodal Test Function F23 with Fixed Dimension](image)

Figure 11i. Convergence Curves of LCGSA and other HAs for Multimodal Test Function F20 with Fixed Dimension

![Convergence Curves of LCGSA and other HAs for Multimodal Test Function F20 with Fixed Dimension](image)
Case (I): $W_1 = w$, $W_2 = 1 - w$

Case (II): $W_1 = 1 - w$, $W_2 = w$

Case (III): $W_1 = \frac{1 - w}{2}$, $W_2 = \frac{1 - w}{2}$

Also, $0 \leq w \leq 1$

The PI graphs of three different weight cases are shown in Figures 6, 7, and 8, respectively. It can be clearly seen in Figures 6, 7, and 8 that the PI of LCGSA is better with respect to GSA in all three weight case categories. Hence, LCGSA is preferred when less execution time and a high success rate is required. In other words, the outcome of PI analysis also demonstrates that the inclusion of chaotic maps and levy flights in GSA increases the convergence rate of LCGSA.
Table 33. Comparison of Results on WBD Problem

| Algorithm | $x_1$    | $x_2$    | $x_3$    | $x_4$    | Best value |
|-----------|----------|----------|----------|----------|------------|
| GSA       | 0.1834   | 4.7518   | 9.5874   | 0.2220   | 2.097      |
| PSO       | 0.7867   | 1.9958   | 2        | 2        | 1.08e+14   |
| BBO       | 1.1183   | 1.3408   | 2        | 2        | 1.08e+14   |
| GA        | 0.9192   | 1.6906   | 2        | 2        | 1.08e+14   |
| DE        | 0.7840   | 2        | 2        | 2        | 1.08e+14   |
| ACO       | 0.5681   | 0.1857   | 1.8312   | 1.6306   | 169163.04  |
| SSA       | 0.13     | 5.22     | 9.06     | 0.21     | 1.7051     |
| SCA       | 0.20     | 3.41     | 9.60     | 0.21     | 1.7693     |
| GWO       | 0.20     | 3.31     | 9.04     | 0.21     | 1.6961     |
| LCGSA1    | 0.27     | 3.00     | 6.87     | 0.36     | 1.8482     |
| LCGSA2    | 0.26     | 3.06     | 7.14     | 0.33     | 1.986      |
| LCGSA3    | 0.24     | 3.72     | 6.00     | 0.47     | 2.1655     |
| LCGSA4    | 0.42     | 1.93     | 6.30     | 0.42     | 2.0336     |
| LCGSA5    | 0.25     | 3.01     | 7.61     | 0.29     | 1.877      |
| LCGSA6    | 0.25     | 2.98     | 7.79     | 0.28     | 1.7997     |
| LCGSA7    | 0.13     | 6.73     | 7.93     | 0.27     | 1.8306     |
| LCGSA8    | 0.19     | 3.52     | 8.89     | 0.21     | 1.74       |
| LCGSA9    | 0.13     | 5.75     | 8.14     | 0.25     | 1.8019     |
| LCGSA10   | 0.39     | 2.03     | 6.55     | 0.39     | 1.871      |

Figure 13. Convergence Analysis of WBD Problem at 500 Iterations
Figure 14. Box Plot Analysis of WBD Engineering Benchmark

Figure 15. Illustration of CSD Problem (Arora, 2000)
Table 34. Comparison of Results on CSD Problem

| Algorithm | $x_1$ | $x_2$ | $x_3$ | Best value  |
|-----------|------|------|------|-------------|
| GSA       | 0.13 | 1.16 | 14.70| 3351.8925   |
| PSO       | 2    | 2    | 2    | 377806.9917 |
| BBO       | 2    | 2    | 2    | 377807.5639 |
| GA        | 2    | 2    | 2    | 377806.9917 |
| DE        | 2    | 2    | 2    | 377806.9917 |
| ACO       | 8.06 | 3.71 | 10.83| 199943.3164 |
| SSA       | 0.14 | 1.18 | 14.11| 3311.3006   |
| SCA       | 0.14 | 1.30 | 11.85| 3311.9238   |
| GWO       | 0.14 | 1.30 | 11.95| 3311.3008   |
| LCGSA1    | 0.14 | 1.30 | 13.62| 3342.1326   |
| LCGSA2    | 0.13 | 1.30 | 11.82| 3366.852    |
| LCGSA3    | 0.14 | 1.19 | 13.95| 3311.3035   |
| LCGSA4    | 0.11 | 0.89 | 13.64| 3346.1557   |
| LCGSA5    | 0.14 | 1.30 | 11.94| 3311.3004   |
| LCGSA6    | 0.11 | 0.99 | 13.82| 3313.1908   |
| LCGSA7    | 0.14 | 1.30 | 13.42| 3314.6356   |
| LCGSA8    | 0.15 | 1.29 | 15.00| 3344.2111   |
| LCGSA9    | 0.11 | 0.84 | 15.00| 3432.1257   |
| LCGSA10   | 0.15 | 1.30 | 13.37| 3368.2782   |

Figure 16. Convergence Analysis of CSD Problem at 500 Iterations
Figure 17. Box plot analysis of CSD Engineering Benchmark

Figure 18. Schematic of PVD Problem (Sandgren, 1990)
Table 35. Comparison of Results on PVD Problem

| Algorithm | $x_1$  | $x_2$  | $x_3$  | $x_4$  | Best value |
|-----------|--------|--------|--------|--------|------------|
| GSA       | 0.96   | 31.31  | 51.03  | 90.36  | 4050.4468  |
| PSO       | 10     | 10     | 52.29  | 74.23  | 204322.1349|
| BBO       | 10     | 10     | 52.77  | 77.80  | 204328.8449|
| GA        | 10     | 10     | 53.84  | 70.54  | 204322.2608|
| DE        | 10     | 10     | 53.66  | 71.73  | 204322.1663|
| ACO       | 120.84 | 39.78  | 173.58 | 128.59 | 1.67e+67   |
| SSA       | 0.97   | 0      | 51.77  | 84.92  | 2577.3415  |
| SCA       | 0      | 0      | 40.36  | 200    | 2585.7947  |
| GWO       | 1.24   | 0      | 65.23  | 10     | 2577.4992  |
| LCGSA1    | 0.93   | 0      | 49.86  | 99.53  | 3934.0113  |
| LCGSA2    | 0.97   | 0      | 51.81  | 84.59  | 3750.8722  |
| LCGSA3    | 0.92   | 0      | 49.23  | 104.54 | 3978.7511  |
| LCGSA4    | 0.88   | 0      | 47.06  | 123.50 | 3992.9148  |
| LCGSA5    | 0.95   | 0      | 50.36  | 95.54  | 3925.897   |
| LCGSA6    | 0.94   | 0      | 49.91  | 99.03  | 3935.6446  |
| LCGSA7    | 0.90   | 0      | 48.09  | 114.25 | 3824.3009  |
| LCGSA8    | 0.91   | 0      | 48.41  | 118.50 | 3755.795   |
| LCGSA9    | 0.97   | 0      | 51.42  | 87.49  | 3872.5813  |
| LCGSA10   | 0.98   | 0      | 52.12  | 82.39  | 3832.4105  |
6.6. Comparison of Simulation Results of LCGSA with other Algorithms

The simulation results of LCGSA have been compared with nine different HAs including standard GSA, classical PSO, BBO, GA, DE, ACO, SSA, SCA, and GWO. Moreover, for a fair comparative analysis, all the algorithms were initialized with the same population size of 50 while 500 iterations were selected as a stopping criterion. Besides, Table 32 presents the average objective function values of different algorithms compared with LCGSA. In fact, it depicts the very competitive results of LCGSA than other participating algorithms.

6.7. Convergence Analysis of LCGSA

The convergence curves of LCGSA, standard GSA, and other participating heuristic algorithms have been presented in Figure(s) 9, 10, and 11 for different benchmark functions. Basically, convergence curves are plotted considering median values of fitness function obtained in independent runs by optimization algorithms. In these curves, X-axis represents iterations and Y-axis depicts objective function values. Moreover, it can be clearly seen that LCGSA shows a better convergence rate as compared to standard GSA. In addition, LCGSA provides competitive and promising results as far as the convergence rate is concerned with respect to other HAs.

7. APPLICATION OF LCGSA FOR ENGINEERING DESIGN OPTIMIZATION

In this section, the proposed Levy-Chaotic GSA has been applied to minimize the design parameter values and the cost of the engineering benchmarks. In fact, three classical engineering design problems including Welded Beam Design (WBD), Compression Spring Design (CSD), and Pressure Vessel Design (PVD) have been chosen to test the applied problem-solving potential of LCGSA. It is obvious that constrained problems consist of both equality and inequality constraints. To deal with the aforementioned constraints, the Penalty Function Method (PFM) has been employed. In PFM, constraints are converted into unconstrained ones so that HA can be used to solve them. The PFM can be mathematically represented in Equation (20).

\[
\text{Min } F(x) = f(x) + \lambda \sum_{n=1}^{K} \max \left(0, g_n\right)
\]  

(20)

Such that \(f(x)\) is the cost function and \(\lambda\) is the penalty coefficient. Moreover, \(g_n\) represents particular constraint of the optimization benchmark and \(K\) is the number of constraints.

7.1. Welded Beam Design (WBD) Problem

The WBD problem is a classic mechanical engineering design benchmark as shown in Figure 12. The objective of the WBD framework is to minimize the fabrication cost of the welded beam. It consists of four variables which include weld thickness (h), weld length (l); thickness (b), and height (t) of the bar. There are number of constraints involved namely buckling load on the bar (\(P_c\)), shear stress (\(\tau\)), beam blending stress (\(\theta\)), side constraints, and end deflection of the beam (\(\delta\)). The mathematical formulation of the WBD problem is as follows:

Consider \(\vec{l} = \begin{bmatrix} l_1 & l_2 & l_3 & l_4 \end{bmatrix} = [h l t b] = [x_1 x_2 x_3 x_4]\)

Minimize \(f(\vec{l}) = l_1 l_2 1.10471 + 0.04811 l_3 l_4 (14.0 + l_2)\)

Subject to \(s_1(\vec{l}) = \tau(\vec{l}) - \tau_{\text{max}} \leq 0\)

\(s_2(\vec{l}) = \sigma(\vec{l}) - \sigma_{\text{max}} \leq 0\)

\(s_3(\vec{l}) = \delta(\vec{l}) - \delta_{\text{max}} \leq 0\)
\[ s_4(\vec{I}) = l_4 - l_3 \leq 0 \]
\[ s_5(\vec{I}) = P - P_c(\vec{I}) \leq 0 \]
\[ s_6(\vec{I}) = 0.125 - l_1 \leq 0 \]
\[ s_7(\vec{I}) = 1.10471l_1^2 + 0.0481l_1^4 (14.0 + l_2) - 5.0 \leq 0 \]

Decision variable interval values

- \[ 0.1 \leq l_4 \leq 2 \]
- \[ 0.1 \leq l_5 \leq 10 \]
- \[ 0.1 \leq l_6 \leq 10 \]
- \[ 0.1 \leq l_7 \leq 2 \]

Where

\[ \tau(\vec{I}) = \sqrt{\tau^2 + 2(\tau^-)(l_2 / R) + (\tau^-)^2} \]
\[ \tau^- = P / \sqrt{2l_1}, \quad \tau^+ = MR / L, \quad M = p (L + l_2 / 2) \]
\[ R = \sqrt{\frac{l_4}{4} + (l_1 + l_3 / 2)^2} \]
\[ J = 2 \left( \sqrt{2l_4} \left[ \left( l_2^2 / 4 \right) + \left( l_1 + l_3 / 2 \right)^2 \right] \right) \]
\[ \sigma(\vec{I}) = 6PL / (l_3 l_4), \quad \delta(\vec{I}) = 6P L^3 / E l_3 l_4 \]
\[ P_c(\vec{I}) = \frac{4.013E \sqrt{l_3 l_4} / 36 \left( 1 - \frac{l_3}{2L} \sqrt{E / 4G} \right)}{L^2} \]

- \[ \sigma_{\max} = 30000 \text{ psi} \]
- \[ G = 12 \times 10^6 \text{ psi} \]
- \[ P = 6000 \text{ lb} \]
- \[ \tau_{\max} = 13600 \text{ psi} \]
- \[ L = 14 \text{ in.} \]
- \[ \delta_{\max} = 0.25 \text{ in.} \]
- \[ E = 3 \times 10^6 \text{ psi} \]

### 7.1.1. Simulation Results of WBD Problem

The WBD problem has been solved using ten different versions of LCGSA and nine other HAs. The simulation results are reported in Table 33. It can be seen that LCGSA provides optimal cost value of the welded beam as compared to GSA. However, GWO also gives a better value of the cost function. Moreover, PSO, BBO, GA, DE, and ACO depict large values of the WBD benchmark.

The convergence analysis among participating algorithms including the best performing version of LCGSA, that is, LCGSA8 is shown in Figure 13. The convergence curves indicate efficient exploitation capability of LCGSA than most of the competing algorithms while GWO, GSA, and SCA also provide promising results. However, PSO, BBO, GA, DE, ACO, and SSA show sub-optimal convergence power indicating issues in handling complex search spaces of the WBD problem.

In addition, the in-depth analysis of the simulation results of the WBD problem has been performed through box plot analysis as shown in Figure 14. It clearly shows that LCGSA has lower values for the median, lower, and upper quartile as compared to standard GSA. Moreover, GWO also provides competitive cost values for the WBD benchmark.

### 7.2. Compression Spring Design (CSD) Problem

The CSD problem is another famous engineering benchmark in which the objective is to minimize the weight of the spring. The schematic of the CSD problem is shown in Figure 15. It consists of three decision variables namely the diameter of the coil (D), the diameter of the wire (d), and a number of the active coils (N).

The mathematical formulation of the CSD problem is as follows:
Consider $\vec{l} = [l_1 l_2 l_3] = [d \, D \, N]$

Minimize $f(\vec{l}) = (l_3^2 + 2) l_2 l_1^2$

Subject to $s_1 (\vec{l}) = 1 - \frac{l_3^3}{717851} \leq 0$

$s_2 (\vec{l}) = \frac{4 l_2^2 - l_3^3}{12566 (l_3 l_2^3 - l_4^3)} + \frac{1}{5108 l_3^2} \leq 0$

$s_3 (\vec{l}) = 1 - \frac{140.45 l_3}{l_2^2 l_3} \leq 0$

$s_4 (\vec{l}) = \frac{l_1 + l_2}{1.5} - 1 \leq 0$

Decision variable interval values

- $0.05 \leq l_1 \leq 2.00$
- $0.25 \leq l_2 \leq 1.30$
- $2.00 \leq l_3 \leq 15.0$

7.2.1. Simulation Results of CSD Problem

Table 34 depicts simulation results of HAs including LCGSA for CSD benchmark. It can be seen that LCGSA5 (3311.3004) has a better objective function value as compared to standard GSA (3351.8925). Besides, SCA, SSA, and GWO also show promising results. However, PSO, BBO, GA, DE, and ACO have sub-optimal values for the CSD engineering benchmark.

Moreover, Figure 16 shows convergence graphs of competing algorithms for the CSD problem. It is clear that LCGSA has a fast convergence rate as compared to most of the algorithms including GWO. Moreover, GSA, SSA, and SCA also depict efficient convergence power. However, PSO, ACO, and BBO have a slow convergence rate for the CSD problem.

The box plots of the CSD problem are presented in Figure 17. It clearly indicates lower median values of LCGSA as compared to standard GSA. Besides, GWO, SCA, and SSA also provide better median values.

7.3. Pressure Vessel Design (PVD) Problem

The goal of the PVD engineering benchmark is to minimize the overall fabrication cost of the cylindrical vessel as shown in Figure 18. It has four parameters including inner radius (R), head thickness ($T_h$), length of the cylindrical shell (L), and thickness of the shell ($T_i$).

Mathematically, the PVD problem can be defined as follows:

Consider $\vec{l} = [l_1 l_2 l_3 l_4] = [T_h \, T_i \, R \, L]$

Minimize $f(\vec{l}) = 0.6224 l_1 l_3^2 l_4^4 + 1.781 l_2 l_3^2 + 3.1661 l_4^2 l_3 + 19.84 l_1^2 l_3$

Subject to $s_1 (\vec{l}) = -l_1 + 0.0193 l_3 \leq 0$

$s_2 (\vec{l}) = -l_3 + 0.00954 l_3 \leq 0$

$s_3 (\vec{l}) = -\pi l_3^2 l_4 + \frac{4}{3} \pi l_3^3 + 1296000 \leq 0$

$s_4 (\vec{l}) = l_4 - 240 \leq 0$

Decision variable interval values

- $0 \leq l_1 \leq 99$
- $0 \leq l_2 \leq 99$
- $0 \leq l_3 \leq 99$
\begin{align*}
10 \leq l_3 & \leq 200 \\
10 \leq l_4 & \leq 200
\end{align*}

### 7.3.1. Simulation Results of PVD Problem

The experimental outcomes of the PVD problem using nine HAs and ten versions of LCGSA are reported in Table 35. The LCGSA has a better objective function value as compared to standard GSA. Moreover, SCA, GWO, and SSA also provide efficient results.

The convergence behavior analysis has been shown in Figure 19. It clearly indicates the high exploitation power of LCGSA than most of the competing algorithms including SCA and SSA. Besides, GWO too shows a high convergence rate indicating appreciable intensification power in getting optimal candidate solutions while dealing with complex and non-linear search spaces.

#### Figure 19. Convergence Analysis of PVD Problem at 500 Iterations

![Convergence Analysis of PVD Problem at 500 Iterations](image)

#### Figure 20. Box plot Analysis of PVD Engineering Benchmark

![Box plot Analysis of PVD Engineering Benchmark](image)
The box plots of the PVD benchmark are shown in Figure 20. It also indicates the optimal performance of LCGSA as compared to GSA in terms of median, minimum, and maximum values. The minimum and maximum values of LCGSA are 3600 and 4200 while GSA has values in the 10e+4 range which indicates less intensification power.

8. CONCLUSION AND FUTURE DIRECTIONS

In this paper, a modified version of GSA has been proposed based on levy flight distribution and chaos theory in order to rectify entrapment in local minima and slow convergence difficulties of standard GSA. In LCGSA, exploration is carried out by levy flight whereas exploitation is performed through the chaotic map(s). The effectiveness of LCGSA is validated by applying it to 23 classical benchmark functions and three engineering design problems.

The experimental and statistical results depict the effective performance of LCGSA over standard GSA. As far as simulation results of 23 benchmark functions are concerned, LCGSA provides optimal performance on 15 test functions whereas standard GSA on 6 benchmark functions only. However, both displayed the same performance on the remaining two benchmark functions. Furthermore, LCGSA also performed well on all of the three engineering design problems in comparison to standard GSA. In addition, LCGSA also showed very competitive results as compared to the other eight peer algorithms. Hence, it can be concluded while considering experimental results of benchmark functions and engineering design problems that LCGSA has been successful in overcoming entrapment in local minima and premature convergence issues of standard GSA. In simpler terms, LCGSA is an effective optimizer in contrast to standard GSA.

However, it is also obvious that LCGSA has some disadvantages that have to be resolved to make it a more efficient and coherent optimizer. Firstly, it is clear from the run time analysis that LCGSA is taking comparatively more run time to find the optimal solutions than standard GSA. So, a new exponential equation for gravitational constant can be formulated to further increase the exploitation rate and consequently, accelerating the convergence speed of LCGSA. Secondly, the LCGSA results for engineering constraints can be further enhanced by utilizing non-linear and complex penalty function methods in order to handle complex solution spaces of engineering benchmarks.

In future studies, it will be worthwhile to apply LCGSA for training MLP neural networks for feature selection. Moreover, multi-objective LCGSA can also be implemented. A binary version of LCGSA is not a bad idea, as it can be utilized to solve a number of binary problems like quadratic assignment problem. In addition, discrete and mixed-integer problems can be solved by utilizing LCGSA. Besides, the applicability of LCGSA for solving high dimensional and complex real-world problems can also be investigated.
Ahmed, A. M., Rashid, T. A., & Saeed, S. A. (2020). Cat Swarm Optimization Algorithm: A Survey and Performance Evaluation. *Computational Intelligence and Neuroscience*, 2020, 1–20. Advance online publication. doi:10.1155/2020/4854895 PMID:32405296

Alatas, B. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, 37(8), 5682–5687. doi:10.1016/j.eswa.2010.02.042

Alatas, B. (2010). Chaotic harmony search algorithms. *Applied Mathematics and Computation*, 216(9), 2687–2699. doi:10.1016/j.amc.2010.03.114

Alatas, B., Akin, E., & Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Solitons & Fractals*, 40(4), 1715–1734. doi:10.1016/j.chaos.2007.09.063

Arora, J. S. (2004). *Introduction to optimum design*. Elsevier. doi:10.1016/B978-012064155-0/50012-4

Basset, M. A., Wang, G. G., Sangaiah, A. K., & Rushdy, E. (2019). Krill herd algorithm based on cuckoo search for solving engineering optimization problems. *Multimedia Tools and Applications*, 78(4), 3861–3884. doi:10.1007/s11042-017-4803-x

Chakraborti, T., & Chatterjee, A. (2014). A novel binary adaptive weight GSA based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns. *Engineering Applications of Artificial Intelligence*, 33, 80–90. doi:10.1016/j.engappai.2014.04.006

Cheng, L., Wu, X. H., & Wang, Y. (2018). Artificial flora (AF) optimization algorithm. *Applied Sciences (Basel, Switzerland)*, 8(3), 329. doi:10.3390/app8030329

Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2), 113–127. doi:10.1016/S0166-3615(99)00046-9

Deep, K., & Thakur, M. (2007). A new crossover operator for real coded genetic algorithms. *Applied Mathematics and Computation*, 188(1), 895–911. doi:10.1016/j.amc.2006.10.047

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3–18. doi:10.1016/j.swevo.2011.02.002

Dorigo, M., & Caro, G. D. (1995). Ant Colony Optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 26, 29–41. doi:10.1109/3477.484436 PMID:18263004

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. 6th IEEE International Symposium on Micro Machine and Human Science (MHS ’95). doi:10.1109/MHS.1995.494215

Eiben, A. E., & Schippers, C. A. (1998). On evolutionary diversification and intensification. *Fundamenta Informaticae*, 35(1-4), 35–50. doi:10.3233/FI-1998-35123403

Gandomi, A. H., & Yang, X. S. (2014). Chaotic Bat Algorithm. *Journal of Computational Science*, 5(2), 224–232. doi:10.1016/j.jocs.2013.10.002

Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 89–98. doi:10.1016/j.cnsns.2012.06.009

Gandomi, A. H., Yun, G. J., Yang, X. S., & Talatahari, S. (2013). Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*, 18(2), 327–340. doi:10.1016/j.cnsns.2012.07.017

Gao, W., Dimitrov, D., & Abdo, H. (2018). Tight independent set neighborhood union condition for fractional critical deleted graphs and ID deleted graphs. *Discrete & Continuous Dynamical Systems-S*, 12, 711-721.

Gao, W., Guirao, J. L., Basavanagoud, B., & Wu, J. (2018). Partial multi-dividing ontology learning algorithm. *Information Sciences*, 467, 35–58. doi:10.1016/j.ins.2018.07.049
Gouthamkumar, N., Sharma, V., & Naresh, R. (2015). Disruption based gravitational search algorithm for short term hydrothermal scheduling. *Expert Systems with Applications, 42*(20), 7000–7011. doi:10.1016/j.eswa.2015.05.017

Güvenç, U., Sümmez, Y., Duman, S., & Yörükeren, N. (2012). Combined economic and emission dispatch solution using gravitational search algorithm. *Scientia Iranica, 19*(6), 1754–1762. doi:10.1016/j.scient.2012.02.030

Hakli, H., & Üğuz, H. (2014). A novel particle swarm optimization algorithm with Levy flight. *Applied Soft Computing, 23*, 333–345. doi:10.1016/j.asoc.2014.06.034

Halliday, D., Resnick, R., & Walker, J. (2000). *Fundamentals of Physics* (6th ed.). Wiley.

Hayyolalam, V., & Kazem, A. A. P. (2020). Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 87*, 103–249. doi:10.1016/j.engappai.2019.103249

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press. doi:10.7551/mitpress/1090.001.0001

Hu, C., Li, Z., Zhou, T., Zhu, A., & Xu, C. (2016). A multi-verse optimizer with levy flights for numerical optimization and its application in test scheduling for network-on-chip. *PLoS One, 11*(12), e0167341. Advance online publication. doi:10.1371/journal.pone.0167341 PMID:27926946

Jensi, R., & Jiji, G. W. (2016). An enhanced particle swarm optimization with levy flight for global optimization. *Applied Soft Computing, 43*, 248–261. doi:10.1016/j.asoc.2016.02.018

Jiang, S., Zhang, C., & Chen, S. (2020). Sequential Hybrid Particle Swarm Optimization and Gravitational Search Algorithm with Dependent Random Coefficients. *Mathematical Problems in Engineering*. Advance online publication. doi:10.1155/2020/1957812

Jothiprakash, V., & Arunkumar, R. (2013). Optimization of hydro-power using evolutionary algorithms coupled with chaos. *Water Resources Management, 27*(7), 1963–1979. doi:10.1007/s11269-013-0265-8

Kang, K., Bae, C., Yeung, H. W. F., & Chung, Y. Y. (2018). A hybrid gravitational search algorithm with swarm intelligence and deep convolutional feature for object tracking optimization. *Applied Soft Computing, 66*, 319–329. doi:10.1016/j.asoc.2018.02.037

Kennedy, J., & Eberhart, R. C. (1993). Particle Swarm Optimization. *International Conference on Neural Networks*, 1942-1948. doi:10.1109/ICNN.1995.488968

Khajoeei, F., & Rashedi, E. (2016). A new version of Gravitational Search Algorithm with negative mass. *2016 1st IEEE Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, 1-5. doi:10.1109/CSIEC.2016.7482123

Khan, T. A., & Ling, S. H. (2020). An improved gravitational search algorithm for solving an electromagnetic design problem. *Journal of Computational Electronics, 19*(2), 1–7. doi:10.1007/s10825-020-01476-8

Li, C., Zhou, J., Xiao, J., & Xiao, H. (2012). Parameters identification of chaotic system by chaotic gravitational search algorithm. *Chaos, Solitons, and Fractals, 45*(4), 539–547. doi:10.1016/j.chaos.2012.02.005

Li, Y., Li, X., Liu, J., & Ruan, X. (2019). An improved bat algorithm based on lévy flights and adjustment factors. *Symmetry, 11*(7), 925. doi:10.3390/sym11070925

Li, Z., Zhou, Y., Zhang, S., & Song, J. (2016). Lévy-flight moth-flame algorithm for function optimization and engineering design problems. *Mathematical Problems in Engineering.*

Mingjun, J., & Huanwen, T. (2004). Application of chaos in simulated annealing. *Chaos, Solitons, and Fractals, 21*(4), 933–941. doi:10.1016/j.chaos.2003.12.032

Mirjalili, S. (2015). The ant lion optimizer. *Advances in Engineering Software, 83*, 80–98. doi:10.1016/j.advengsoft.2015.01.010

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133. doi:10.1016/j.knosys.2015.12.022
Mirjalili, S., & Gandomi, A. H. (2017). Chaotic gravitational constants for the gravitational search algorithm. *Applied Soft Computing, 53*, 407–419. doi:10.1016/j.asoc.2017.01.008

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163–191. doi:10.1016/j.advengsoft.2017.07.002

Mirjalili, S., Hashim, S. Z. M., & Sardroudi, H. M. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation, 218*(22), 11125–11137. doi:10.1016/j.amc.2012.04.069

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67. doi:10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Pelusi, D., Mascella, R., & Tallini, L. (2018). A fuzzy gravitational search algorithm to design optimal IIR filters. *Energies, 11*(4), 736. doi:10.3390/en11040736

Peng, Z., Dong, K., Yin, H., & Bai, Y. (2018). Modification of Fish Swarm Algorithm Based on Levy Flight and Firefly Behavior. *Computational Intelligence and Neuroscience, 2018*, 1–13. Advance online publication. doi:10.1155/2018/9827372 PMID:30344601

Prajna, K., Rao, G. S. B., Reddy, K. V. V. S., & Maheswari, R. U. (2014). A new approach to dual channel speech enhancement based on gravitational search algorithm (GSA). *International Journal of Speech Technology, 17*(4), 341–351. doi:10.1007/s10772-014-9232-x

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences, 179*(13), 2232–2248. doi:10.1016/j.ins.2009.03.004

Rather, S. A., & Bala, P. S. (2019a). A holistic review on gravitational search algorithm and its hybridization with other algorithms. *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 1-6. 2019 IEEE. doi:10.1109/ICECCT.2019.8869279

Rather, S. A., & Bala, P. S. (2019b). Analysis of gravitation based optimization algorithms for clustering and classification. In *Handbook of research on big data clustering and machine learning* (pp. 77–99). IGI Global. doi:10.4018/978-1-7998-0106-1.ch005

Rather, S. A., & Bala, P. S. (2019c). Hybridization of constriction coefficient based Particle Swarm Optimization and Gravitational Search Algorithm for function optimization. *2019 Elsevier International Conference on Advances in Electronics, Electrical, and Computational Intelligence (ICAEEC-2019)*, 1-6. 2019 Elsevier. doi:10.2139/ssrn.3576489

Rather, S. A., & Bala, P. S. (2019d). Hybridization of Constriction Coefficient-Based Particle Swarm Optimization and Chaotic Gravitational Search Algorithm for Solving Engineering Design Problems. In *International Conference on Advanced Communication and Networking* (pp. 95-115). Springer. doi:10.1007/978-981-15-3852-0_7

Rather, S. A., & Bala, P. S. (2020a). A Hybrid Constriction Coefficient Based Particle Swarm Optimization and Gravitational Search Algorithm for Training Multi-Layer Perceptron (MLP). *International Journal of Intelligent Computing and Cybernetics, 13*(2), 129–165. doi:10.1108/IJICC-09-2019-0105

Rather, S. A., & Bala, P. S. (2020b). Swarm-based chaotic gravitational search algorithm for solving mechanical engineering design problems. *World Journal of Engineering, 17*(1), 97–114. doi:10.1108/WJE-09-2019-0254

Rather, S. A., & Bala, P. S. (2021a). Application of Constriction Coefficient Based Particle Swarm Optimization and Gravitational Search Algorithm for Solving Practical Engineering Design Problems. *International Journal of Bio-inspired Computation, 17*(4), 246–259. doi:10.1504/IJIBC.2021.116617

Rather, S. A., & Bala, P. S. (2021b). Constriction Coefficient Based Particle Swarm Optimization and Gravitational Search Algorithm for Multilevel Image Thresholding. In *Expert Systems: International Journal of Knowledge Engineering and Neural Networks, 38*(7). Advance online publication. doi:10.1111/exsy.12717

Rodenas, R. G., Linares, L. J., & Gomez, J. A. L. (2019). A memetic chaotic gravitational search algorithm for unconstrained global optimization problems. *Applied Soft Computing, 79*, 14–29. doi:10.1016/j.asoc.2019.03.011
Sajedi, H., & Razavi, S. F. (2017). DGSA: Discrete gravitational search algorithm for solving knapsack problem. *Operations Research, 17*(2), 563–591. doi:10.1007/s12351-016-0240-2

Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design, 112*(2), 223–229. doi:10.1115/1.2912596

Saremi, S., Mirjalili, S., & Lewis, A. (2014). Biogeography-based optimization with chaos. *Neural Computing & Applications, 25*(5), 1077–1097. doi:10.1007/s00521-014-1597-x

Shabani, A., Asgarian, B., Gharebaghi, S. A., Salido, M. A., & Giret, A. (2019). A new optimization algorithm based on search and rescue operations. *Mathematical Problems in Engineering, 2019*, 1–23. Advance online publication. doi:10.1155/2019/2482543

Shan, X., Liu, K., & Sun, P. L. (2016). Modified bat algorithm based on lévy flight and opposition based learning. *Scientific Programming, 2016*, 1–13. Advance online publication. doi:10.1155/2016/8031560

Shen, D., Jiang, T., Chen, W., Shi, Q., & Gao, S. (2015). Improved chaotic gravitational search algorithms for global optimization. *IEEE Congress on Evolutionary Computation (EC), 2015*, 1220-1226. doi:10.1109/CEC.2015.7257028

Simon, D. (2008). Biogeography Based Optimization. *IEEE Transactions on Evolutionary Computation, 12*(6), 702–713. doi:10.1109/TEVC.2008.919004

Storn, R. (1995). *Differential Evolution*. Technical report TR-95-012, International Computer Science Institute.

Sulaiman, M. H., Mustaffa, Z., Saari, M. M., & Daniyal, H. (2020). Barnacles Mating Optimizer: A new bio-inspired algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 87*, 103–330. doi:10.1016/j.engappai.2019.103330

Talatahari, S., Azar, B. F., Sheikholeslami, R., & Gandomi, A. (2012). Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation, 17*(3), 1312–1319. doi:10.1016/j.cnsns.2011.08.021

Wang, L., & Zhong, Y. (2015). Cuckoo search algorithm with chaotic maps. *Mathematical Problems in Engineering*. Advance online publication. doi:10.1155/2015/715635

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82. doi:10.1109/4235.585893

Xie, J., Zhou, Y., & Chen, H. (2013). A novel bat algorithm based on differential operator and Lévy flights trajectory. *Computational Intelligence and Neuroscience, 2013*, 1–13. Advance online publication. doi:10.1155/2013/453812 PMID:23606827

Yang, X. S. (2010). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press.

Yao, J. F., Mei, C., Peng, X. Q., Hu, Z. K., & Hu, J. (2001). A new optimization approach-chaos genetic algorithm. *Systems Engineering, 1*, 0-15.

Yazdani, S., Nezamabadi-pour, H., & Kamyab, S. (2014). A gravitational search algorithm for multimodal optimization. *Swarm and Evolutionary Computation, 14*, 1–14. doi:10.1016/j.swevo.2013.08.001

Yin, M., Hu, Y., Yang, F., Li, X., & Gu, W. (2011). A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering. *Expert Systems with Applications, 38*(8), 9319–9324. doi:10.1016/j.eswa.2011.01.018

Zandevakili, H., Rashedi, E., & Mahani, A. (2019). Gravitational search algorithm with both attractive and repulsive forces. *Soft Computing, 23*(3), 783–825. doi:10.1007/s00500-017-2785-2

Zhenyu, G., Bo, C., Min, Y., & Binggang, C. (2006). Self-adaptive chaos differential evolution. In *International Conference on Natural Computation: Advances in Natural Computation, 972-975*. doi:10.1007/11881070_128
APPENDIX A – NOMENCLATURE

Figure 21. Nomenclature

| Algorithm | Extension                                      |
|-----------|------------------------------------------------|
| HA        | Heuristic Algorithm                            |
| GSA       | Gravitational Search Algorithm                 |
| PSO       | Particle Swarm Optimization                    |
| LCGSA     | Levy Flight and Chaos Theory based GSA         |
| GA        | Genetic Algorithm                              |
| DE        | Differential Evolution                         |
| ACO       | Ant Colony Optimization                        |
| BBO       | Biogeography Based Optimization                |
| GWO       | Grey Wolf Optimizer                            |
| ALO       | Ant Lion Optimizer                             |
| SCA       | Sine-Cosine Algorithm                          |
| SSA       | Salp Swarm Algorithm                           |
| AFO       | Artificial Flora Optimization Algorithm        |
| SRO       | Search and Rescue Operations Optimization      |
| CSO       | Cat Swarm Optimization                         |
| BWO       | Black Widow Optimization                       |
| BMO       | Barnacles Mating Optimizer                     |
| BA        | Bat Algorithm                                  |
| WBD       | Welded Beam Design Problem                     |
| CSD       | Compression Spring Design Problem              |
| PVD       | Pressure Vessel Design Problem                 |

Sajad Ahmad Rather completed his Bachelor of Technology (B.Tech., 2014) in Computer Science and Engineering (CSE) from Islamic University of Science and Technology (IUST), Awantipora, Kashmir, and Master of Technology (M.Tech., 2016) degree in CSE from the SRM University, Chennai, India; both with a distinction. Currently, he is pursuing his doctorate in Computer Science and Engineering at Pondicherry University, India. He has publications in reputed international journals and presented his research in national and international conferences (Springer, Elsevier, ACM, IEEE, etc.). He is a member of IEEE, ACM, ISTE, IEI, IETE, and IAENG. Moreover, he has authored three books. He is also a reviewer of the IGI International Journal of Applied Metaheuristic Computing (IJAMC) and the International Journal of Intelligent Computing and Cybernetics (IJICCC) Emerald. His research areas of interest include Artificial Intelligence (AI), Swarm Intelligence, Machine learning (ML), Meta-heuristics, Image Segmentation, MLP Training, and Engineering Optimization.

P. Shanthi Bala is working as an Assistant Professor in the Department of Computer Science, School of Engineering and Technology, Pondicherry University, India. She has published papers in peer-reviewed and SCI International journals. Her research interests include Artificial Intelligence, Machine Learning, Neural Networks, Internet of Things (IoT), Deep Learning, Distributed Computing, and Ontology.