Hybridization of Chaotic Maps and Gravitational Search Algorithm for Constrained Mechanical and Civil Engineering Design Frameworks: CGSA for Mechanical and Civil Engineering Design Optimization

Sajad Ahmad Rather, Pondicherry University*
https://orcid.org/0000-0002-0351-7705
P. Shanthi Bala, Pondicherry University
https://orcid.org/0000-0003-0576-4424

ABSTRACT

The chaotic gravitational search algorithm (CGSA) is a physics-based heuristic algorithm inspired by Newton's law of universal gravitation. It uses 10 chaotic maps for optimal global search and fast convergence rate. The advantages of CGSA have been incorporated in various mechanical and civil engineering design frameworks which include speed reducer design (SRD), gear train design (GTD), three bar truss design (TBTD), stepped cantilever beam design (SCBD), multiple disc clutch brake design (MDCBD), and hydrodynamic thrust bearing design (HTBD). The CGSA has been compared with 11 state-of-the-art stochastic algorithms. In addition, a non-parametric statistical test, namely the signed Wilcoxon rank-sum test, has been carried out at a 5% significance level to statistically validate the results. The simulation results indicate that CGSA shows efficient performance in terms of high convergence speed and minimization of the design parameter values as compared to other heuristic algorithms. The source codes are publicly available on Github (i.e., https://github.com/SajadAHMAD1).

KEYWORDS
Chaotic Gravitational Search Algorithm (CGSA), Chaotic Maps, Engineering Design, Exploitation, Exploration, Hybridization, Nature-Inspired Computing

1. INTRODUCTION

In the last decade, heuristic optimization algorithms are creating ripples in the international computational intelligence community of researchers. The research works and the applications of heuristic algorithms in various fields have been phenomenal. Heuristic algorithms are getting popular due to their stochastic nature and simplicity. According to the No Free Lunch theorem (NFLT) (Wolpert and Macready, 1997), a single algorithm cannot solve all optimization problems. In simpler words, if an optimization algorithm solves some problems with high performance, there is a high probability...
that it performs badly in solving another type(s) of the optimization problem(s). Hence, the researchers have invented many optimization algorithms and every year new algorithms are being proposed. The famous optimization algorithms include Particle Swarm Optimization (PSO) (Kennedy et al., 1993) which is inspired by the social behavior of birds and fishes, Ant Colony Optimization (ACO) (Dorigo et al., 1996) based on the searching behavior of ants, Biogeography Based Optimization (BBO) (Simon, 2008) inspired from the distribution and migration models of species, Differential Evolution (DE) (Storn and Price, 1995) and Genetic Algorithm (GA) (Tang et al., 1996) which are motivated by the theory of evolution. The heuristic Algorithms (HAs) have been applied to solve various problems of computer science and other fields of study such as electronics, biology, oil industry, and so on.

In computer science, HAs have been utilized for function optimization (Du and Li, 2008; Yao et al., 1999), control objectives (Baojiang and Shiyong, 2007; Karakuza, 2008; Kim et al., 2008), pattern recognition (Liu et al., 2008; Tan and Bhanu, 2006), modeling of filters (Kalinlia and Karabogab, 2005) and optimal processing of images (Cordon et al., 2006; Nezamabadi-pour et al., 2006). In electronic science, load dispatch (Beigvand et al., 2016; 2017) and optimal power flow (Bhowmiket al., 2015) problems have been efficiently solved by optimization algorithms.

All optimization algorithms consist of a random population of agents that are used for finding the candidate solutions in the search space. The process starts with the initialization of agents in the search space. Then, the algorithm goes through many iterations and each independent trial gives the feasible candidate solutions until the end of the criterion is met. The best feasible candidate solution from all the iterations is selected as the optimal solution. It is quite amazing that stochastic algorithms consist of a few fundamental steps such as exploration and exploitation. Exploration consists of the search space of the algorithm. During this phase, the candidate solutions go through a number of changes. Moreover, Exploitation is the capability of finding global optima around different feasible solutions. The candidate solutions face small changes during the exploitation phase. It has been seen that if an optimization algorithm has good exploration capability, then it will be lacking in good exploitation capability and vice versa (Eiben and Schipper, 1998). Hence, these are inversely proportional to each other. Previously, researchers were using random walks and gradient descent methods for improving exploration and exploitation, respectively. However, they have the drawback of increasing the overall computational cost of the algorithm. In the last decade, researchers are now utilizing chaotic maps for increasing diversification and local exploitation of search space in order to find the optimal candidate solutions (Mirjalili et al., 2017; Gandomi et al., 2012).

Chaos theory is the study of dynamic systems. The interesting property of these systems is that when there is a minor change in the system, the whole system gets affected. In simpler words, change in the initial parameter(s) creates variations throughout the system. Moreover, randomness is not necessary for chaotic systems rather deterministic systems also show chaotic behavior (Kellert, 2017). The optimization algorithms utilize Initial Parameter Sensibility (IPS) property of chaotic maps for fast exploitation, global exploration, and alleviation from local minima entrapment problem.

The meta-heuristic algorithms show good results on benchmark functions but they perform poorly on real-world problems. Besides solving a practical problem is an actual test for checking the application potential of an optimization algorithm. The researchers have used various optimization techniques such as GA, PSO, DE, GSA, BBO, etc. for solving practical problems in other fields. Most of the algorithms are evolutionary in nature but GSA is a recent meta-heuristic algorithm that is based on physics. The GSA (Rashedi et al., 2009) is a HA that is inspired by Newton's law of universal gravitation and laws of motion. In this work, ten chaotic maps are added to standard GSA and have been applied to six engineering design problems for studying hybridization potential and capability of handling complex numerical ranges of GSA. All six problems have unknown search spaces and are ideal for checking the problem-solving capability of the algorithm. Further, CGSA will be examined for constraint handling and parameter optimization as far as complex engineering benchmarks are concerned.
The remaining article is structured as follows. The motivation for work is explained in section 2. Section 3 provides a literature review of chaotic maps and CGSA. Section 4 describes 10 chaotic maps that are used in the paper. The CGSA is explained in section 5. The experimental analysis is covered in section 6. Finally, section 7 concludes the work.

2. MOTIVATION

The chaotic maps are very sensitive to small changes in input and depict abrupt vibrations in the output. Yao et al. in 2001 have started using chaotic maps in intelligent techniques. They integrated chaotic maps with a genetic algorithm for increasing its exploitation rate. In this paper, authors have used chaos theory as a performance amplification tool with standard GSA for solving engineering design problems.

Further, researchers (Gandomi et al., 2012; Gandomi, 2013) have found that random number sequence generators used for stochastic initialization of agents in search space can be replaced with chaotic sequences. It is because chaotic maps have the properties of stochasticity, ergodicity, and nonlinearity which are ideal for overcoming convergence and local minima stagnation issues of optimization algorithms.

Moreover, there are very few papers in research literature were chaotic maps were employed for solving practical engineering problems (Basset et al., 2019; Shen, 2015). Further, the researchers like Javidi et al., 2006 and wang et al., 2016 have hybridized chaotic maps with the state of the art heuristic algorithms like GA, PSO, BA, CS, etc as shown in Fig. 1. However, they have used only logistic, Gauss, and sinusoidal maps more rigorously than other maps whereas this study employs 10 different chaotic maps for optimization of engineering design frameworks.

In this paper, chaotic maps have been utilized to take gravitational agents from simple search spaces of continuous problems to unknown search spaces of practical constrained engineering design problems. It will be quite interesting to see how chaotic maps can handle complex inequalities of design frameworks.

3. LITERATURE SURVEY

The chaotic maps have unstable and ergodic motions. These non-periodic motions can be converted into sequences and utilized in place of random number generators. In PSO, 8 chaotic maps have been used for parameter adaptation. The chaotic sequences have been utilized for the initialization of particles in the search space. It has been identified that the chaotic maps were successful in increasing the solution quality and global searching capability of PSO (Alatas et al., 2009).

Similarly, Artificial Bee Colony (ABC) is the newest intelligent technique suitable for global optimization. It has been seen that ABC has the drawbacks of slow convergence speed and stagnation in local minima. Seven chaotic maps such as gauss, sinusoidal, sinus, henon, logistic, circle, and tent maps were utilized for producing non-periodic number sequences. The experimental results have shown that two new proposed versions of ABC namely CABC 2 and CABC3 based on chaos theory showed efficient performance than standard ABC for parameter adaptation (Alatas, 2010).

Harmony Search (HS) algorithm is an amazing optimization algorithm inspired by the music player and having characteristics that are absent in other population algorithms like divergence-free property, multi-modal adaptability, and mathematical simplicity. But it has the drawback of slow convergence speed. To overcome the drawback, chaotic maps have been embedded in HS for increasing its local searching capability (Alatas, 2010). Recently, researchers (Kumar et al., 2019) have introduced new HA inspired by mathematical science namely Spherical Search (SS). It is based on the concept of spherical projection where searcher agents are considered as the points on the spherical surface. Besides, to evaluate the performance of SS, thirty constrained problems were utilized. Moreover, the simulation results were compared with other state of the art HAs. Meanwhile,
an interesting optimization technique has been invented by Mohamed et al. (2019a) based on the human experience and wisdom called as Gaining and Sharing Knowledge (GSK) based algorithm. It consists of human knowledge at junior and senior levels. Besides, the GSK has been applied to CEC 2011 and CEC 2017 test functions to benchmark its performance. In addition, ten HAs were utilized for comparative analysis.

Moreover, the chaos theory concepts have also been applied to the genetic algorithm for overcoming its premature convergence problem. The complex and non-periodic chaotic sequences of logistic and tent maps were utilized in place of [0,1] random number sequences. It has been proven that CGA reached to the optimal solution in lesser iterations as compared to standard GA (Javidi and Hosseinpourfard, 2006).

Likewise, Differential Evolution (DE) (Storn and Price, 1995) is one of the intelligent techniques having the global searching capability but it has issues with the exploitation of the search space for finding the candidate solutions. A new version of DE has been proposed namely self-adaptive chaotic DE (SACDE) which has chaotic behavior. The SACDE modifies mutation and crossover operators of DE to increase the local searching capability of standard DE. Two mathematical benchmarks were utilized for testing the performance of DE and SACDE. The SACDE shows optimal performance than the classical DE (Guo et al., 2016). It has been proven that SHADE and LSHADE are two efficient algorithms for solving numerical and constrained optimization problems. However, they faced decreased solution quality and slow convergence drawbacks while dealing with dimensional and complex search spaces. Therefore, mutation schemes have been introduced in SHADE and LSHADE algorithms to resolve the aforementioned issues (Mohamed et al. (2019b)). In addition, the Adaptive Guided DE algorithm (AGDE) is a new variant of DE for getting the balance between intensification and diversification phases. Besides, different mathematical rules were created to accelerate the convergence rate and reduce sensitivity in initialization. Moreover, the effectiveness of AGDE was checked by applying it to the CEC2013 test functions (Mohamed et al., 2019c).

Chaotic initialization and sequences have been introduced in Simulated Annealing (SA) for increasing its speed of getting optimal candidate solutions. Normally, SA uses the Gaussian distribution of random numbers for exploitation but it shows slow convergence speed. The chaotic SA improves the performance of standard SA by taking candidate solutions away from stagnant local optima locations and hence, increasing overall convergence rate (Ming Jun and Haunwan, 2004).

Most of the swarm-based algorithms are inspired by natural phenomena. Likewise, the Firefly Algorithm (FA) is inspired by the aggregation behavior of fireflies. They have two main features namely light flashing and attractiveness. These two factors were embedded with chaotic maps for global search (Gandomi et al., 2013).

Meta-heuristics is becoming a cross disciplinarily field as newly invented algorithms utilizing the concepts such as evolution, gravitation, quantum, annealing, etc. from other subjects. The Krill Herd (KH) is a new algorithm added into the meta-heuristic family based on the movement, diffusion, and motion of sea krill. It has good exploitation power but due to its heavy parallelization behavior, it lacks the fast global searching capability. The stochastic and nonlinearity properties of chaotic maps were utilized for increasing its randomness i.e. diversification (Gai et al., 2014).

Biogeography Based Optimization (BBO) is an intelligent technique inspired by the distribution of species in a particular habitat. It has good diversification power due to its migration operator. However, it has two main issues of slow exploitation and stagnation in local minima. To alleviate the difficulties, 12 chaotic maps were utilized. It was seen that the gauss map showed optimal performance on ten benchmark functions (Saremi et al., 2014).

The imperialist Competitive Algorithm (ICA) is inspired by sociology and political science. In ICA, agents are countries (colonies or imperialist). The chaotic maps were considered for increasing the performance of movement operator as it is necessary for convergence of feasible solutions towards global optima (Talatahari et al., 2012).
Mobility and diversity of candidate solutions in the search space are two main issues in most heuristic algorithms. Meanwhile, Bat Algorithm (BA) has been combined with chaotic maps to improve the randomized behavior of different BA operators. Various mathematical benchmark functions were used for evaluating the performance of CBA and standard BA (Gandomi and Yang, 2014). In another research, Bansal et al. (2017a, 2017b) employed a multi-objective version of BA for reducing the crosstalk in the multiplexing systems. Firstly, Golomb Ruler Sequences (GRS) are created in less computational time. Secondly, GRSs are utilized for channel allocation in the optical wavelength multiplexing systems. The performance metrics like ruler length and wavelength depicted the efficiency and robustness of the multi-objective BA. Likewise, improved multi-objective Big Bang Crunch (BBC) has been utilized for network channel optimization. First of all, BBC is employed to generate the Optimal GRS in minimal computational time. At the same time, BBC based Optimal GRS are then applied to wavelength based multiplexing systems to reduce the crosstalk. The experimental results depicted better performance of the BBC in terms of bandwidth and ruler length (Bansal et al., 2018; Bansal, 2018).

In the same way heuristic algorithms are being employed to tune the parameters of fuzzy controllers. It has been reported that GSA, GWO, charged system search and PSO showed optimal results in the position control of fuzzy systems (Precup and David, 2019). Moreover, it has been seen that HAs are efficient when applied to complex optimization tasks. Besides, the channel allocation problem in computer networking is one such problem where conventional optimization techniques face difficulties. Therefore, five HAs including BBC, Firefly Algorithm (FA), BA, CS, and Flower Pollination Algorithm (FPA) were considered for generating the OGRs in order to provide higher bandwidth to complex multiplexing systems (Bansal (2019)).

The chaos theory concepts have been utilized in Cuckoo search (CS) for increasing its local searching capability. The elitism strategy has been employed for maintaining optimal candidate solutions. Twenty-seven benchmark functions were used for performance evaluation (Wang et al., 2016). In addition, polynomial mutation operators were incorporated into the traditional CS algorithm to increase agent diversity whilst avoiding sub-optimal candidate solutions. Different standard numerical benchmarks were utilized for performance investigation including Rastrigin function, Rosebrock function, Griewank’s function, etc. (Basset et al., 2019).

Gravitational search algorithm (GSA) is inspired by the Newton’s gravitation concept in which agents are in the form of masses. The GSA has good exploration power but it gets stuck in the last iterations that cause a slower convergence rate. The chaotic maps were introduced in GSA using adaptive normalization. The experimental results indicate that the sinusoidal map performs better as compared to the other 9 maps. Further, the chaotic version of GSA has been tested on 12 benchmark functions (Miralalili and Gandomi, 2017).

Five chaotic maps namely logistic, piecewise linear, gauss, sinusoidal, and sinus maps were incorporated into GSA for alleviating its exploitation and trapping in local minima problem. It has been revealed that all maps performed well as compared to standard GSA on six widely used benchmark functions (Shen et al., 2015).

Over the past few years, researchers have proposed many GSA variants such as disruption, quantum, binary, multi-objective, etc. Likewise, Kbest is another variant of GSA that is employed for increasing convergence speed of GSA. The Kbest GSA has the property that its value decreases linearly with every passing iteration. Kbest feature has been embedded with chaotic maps to enhance the exploitation power of GSA (Mittal et al., 2016).

Chaotic maps have been combined with standard GSA for solving the Unit Commitment (UC) problem in electrical power production. The chaotic sequences were used in place of random number generators to initiate the population for discrete optimization. The chaotic GSA has been able to solve six and ten UC problems efficiently as compared to quadratic and selective pruning methods (Li et al., 2016).
It has been seen that GSA has applications in many fields of study especially electronics and electrical sciences where it is employed for designing infinite impulsive response (IIR) filter to increase the band amplification of the signal (Suman et al., 2014). Likewise, GSA variant CKbest GSA is being utilized for image segmentation of RGB-D type of images. It has been hybridized with fuzzy clustering (FC). First, CKBest GSA forms multiple clusters of agents. Then, the hybrid model aggregates all clusters for segmentation. The hybrid model has been applied to the NYUD2 dataset and compared with standard GSA, Fuzzy c-means, and c-means clustering algorithms (Mittal et al., 2018).

Researchers have applied binary CGSA (BCGSA) for band selection in hyper-spectral images to remove the anomalous band information. The experimental results indicated that BCGSA has been successful in reducing the curse of dimensionality problem. Besides, it has shown optimal performance as compared to BPSO, GA, and BCS algorithms (Wang et al., 2018). Furthermore, CGSA has been combined with the Quasi-Newton method, a gradient-based mechanism. It has been seen that GSA has always issues with convergence rate at last iterations. To alleviate the issue, the quasi-newton method acts as a local searching operator in this amalgamation pair. The hybridized algorithm has been tested on 45 benchmark functions and results were promising (Rodenas et al., June 2019).

The literature survey as shown in Table 1 clearly conveys that stochasticity, ergodicity, and non-periodic complex behaviors of chaotic maps have huge potential to solve problems in stochastic algorithms. Besides, chaotic maps have been successfully employed to tackle different practical real-life problems such as band selection, band amplification, and cluster segmentation. Hence, in this paper chaotic GSA has been applied to various mechanical and civil engineering design problems to initiate the application scope of chaotic maps in the engineering field.

| Literature Reference       | Year | Heuristic Algorithm | Chaotic Maps Used | Goal of Hybridization                      |
|----------------------------|------|---------------------|-------------------|--------------------------------------------|
| Ming Jun and Hauwuan       | 2004 | SA                  | Yes               | Resolving local minima problem             |
| Javidi and Hosseinpourfarad| 2006 | GA                  | Yes               | To overcome premature convergence          |
| Alatas et al.              | 2009 | PSO                 | Yes               | Parameter adaptation                       |
| Alatas et al.              | 2010 | ABC                 | Yes               | To enhance convergence                      |
| Alatas et al.              | 2010 | HS                  | Yes               | To increase exploitation rate              |
| Talatatari et al.          | 2012 | ICA                 | Yes               | Convergence speed enhancement              |
| Gandomi et al.             | 2013 | GA                  | Yes               | Global search                              |
| Bai et al.                 | 2014 | KHB                 | Yes               | Diversification of search space            |
| Saremi et al.              | 2014 | BBO                 | Yes               | Benchmark analysis                         |
| Suman et al.               | 2014 | GSA                 | Yes               | Band amplification                         |
| Gandomi and Yang           | 2014 | BA                  | Yes               | Diversified randomization                  |
| Shen et al.                | 2015 | GSA                 | Yes               | Benchmark analysis                         |
| Guo et al.                 | 2016 | DE                  | Yes               | To increase convergence speed              |
| Wang et al.                | 2016 | CS                  | Yes               | Enhanced intensification                    |
| Mittal et al.              | 2016 | KBestGSA            | Yes               | Resolve premature convergence              |
| Li et al.                  | 2016 | GSA                 | Yes               | To solve unit commitment problem           |
| Mirjalili and Gandomi      | 2017 | GSA                 | Yes               | Convergence speed analysis                  |
| Bansal et al.              | 2017 | BA                  | No                | Reduction of crosstalk in multiplexing systems |
| Bansal et al.              | 2018 | BBC                 | No                | Channel allocation in multiplexing systems  |
| Mittal et al.              | 2018 | CKBESTGSA           | Yes               | Cluster segmentation                        |
| Wang et al.                | 2018 | CGSA                | Yes               | Band selection                             |
| Kumar et al.               | 2019 | SS                  | No                | Constrained optimization                    |
| Mohamed et al.             | 2019 | GSK                 | No                | Function optimization                       |
| Mohamed et al.             | 2019 | SHADE/LSHADE        | No                | Performance improvement                     |
| Mohamed et al.             | 2019 | AGDE                | No                | Exploration and exploitation tradeoff       |
| Bansal et al.              | 2019 | BA, CS, …           | No                | Channel allocation in multiplexing systems  |
| Precup and David           | 2019 | GSA, …              | No                | Parameter tuning in the fuzzy controller    |
| Basset et al.              | 2019 | CS                  | Yes               | Global minima investigation                 |
| Rodenas et al.             | 2019 | CGSA-QN             | Yes               | To resolve stagnation in local minima      |
4. CHAOTIC MAPS

As already stated, chaotic maps have stochastic behavior and non-linear motions. The chaotic behavior is not only shown by random systems but also depicted by deterministic systems (Rather et al. 2020b) as it is quite evident from Table 2. Ten chaotic maps have been used in this paper. The chaotic map equations are deterministic but have randomized fluctuations as shown in Figure 1.

Table 2. Chaotic maps embedded with GSA

| S. No. | Chaotic Function | Chaotic Map | Limits |
|--------|-----------------|-------------|--------|
| 1.     | Chebyshev       | \( l_{i+1} = \cos (\delta \cos^{-1}(l_i)) \) | (1,-1) |
| 2.     | Circle          | \( l_{i+1} = \text{mod} (l_i + \frac{\pi}{2}, 1) \) | (0,1) |
| 3.     | Gauss           | \( l_{i+1} = \begin{cases} -l_i, & l_i = 0 \\ \frac{1}{\text{mod}(i,3)}, & \text{otherwise} \end{cases} \) | (0,1) |
| 4.     | Iterative       | \( l_{i+1} = \sin \left(\frac{\pi}{l_i}\right), p=0.7 \) | (-1,1) |
| 5.     | Logistic        | \( l_{i+1} = p \cdot l_i (1 - l_i), p=4 \) | (0,1) |
| 6.     | Piecewise       | \( l_{i+1} = \begin{cases} \frac{l_i}{r}, & 0 \leq l_i \leq 0 \\ \frac{l_i + r}{1 + r}, & 0.5 \leq l_i \leq 0.5 \\ \frac{1 - r}{l_i}, & 1 - r \leq l_i \leq 1 \end{cases}, p=0.4 \) | (0,1) |
| 7.     | Sine            | \( l_{i+1} = \frac{\pi}{k} \sin (\pi l_i), p=4 \) | (0,1) |
| 8.     | Singer          | \( l_{i+1} = x (7.86 l_i - 23.31 l_i^2 + 28.75 l_i^3 - 13.302875 l_i^4), x=1.07 \) | (0,1) |
| 9.     | Sinusoidal      | \( l_{i+1} = p \cdot l_i^2 \sin (\pi l_i), p=2.3 \) | (0,1) |
| 10.    | Tent            | \( l_{i+1} = \begin{cases} \frac{l_i}{0.7}, & l_i \leq 0.7 \\ \frac{10}{3} (1 - l_i), & l_i \geq 0.7 \end{cases} \) | (0,1) |

All chaotic maps have their range of values for initialization, that is, (0,1) and (-1,1). Also, \( l_{i+1} \) denotes X coordinate of the chaotic map while i is the index of the map having an initial value of zero. It should be noted that initial values have significant effects on the output of chaotic maps. So, proper selection of initial value is important for efficient stochastic behavior. For this study, chaotic maps will have the same initial values which were used in Gandomi et al. (2012a, 2012b).

In this study, chaotic maps were embedded in gravitational constant \((G(t))\) of GSA. The gravitational constant is important for a proper balance between intensification and diversification phases. By incorporating chaotic maps in GSA, its exploitation and convergence speed will be enhanced.

5. CHAOTIC GRAVITATIONAL SEARCH ALGORITHM (CGSA)

According to modern physics, nature is composed of four forces such as the gravitational force, the strong nuclear force, the electromagnetic force, and the weak nuclear force. As per classical Newtonian
mechanics, gravitational law is stated as “the gravitational force between two masses is directly proportional to the product of their masses and inversely proportional to the square of the distance between them” (Rather et al., 2017, 2020a; Halliday et al., 2000; Rashedi et al., 2009).

Let \( X = \{x_1, x_2, x_3, \ldots, x_n\} \) be a system with ‘n’ agents, such that \( x_i \in \mathbb{R} \). The force between mass \( i \) and \( j \) is shown in Equation (1).

\[
F_{ij} = G(t) \frac{m_{pi}(t) m_{aj}(t)}{R_{ij}(t)} \left( x_j^d(t) + x_i^d(t) \right)
\]  

(1)

Here, \( m_{pi}(t) \) is passive gravitational mass and \( m_{aj}(t) \) is active gravitational mass. \( R_{ij}(t) \) is the Euclidean distance and \( \varepsilon \) is a positive constant in \( d^n \) dimensional space.

The gravitational constant ‘G’ is important for adjusting the accuracy of the search and is given by Equation (2).

\[
G(t) = G(t_0) e^{-\frac{C_I}{M_I}}
\]

(2)

Where \( G(t) \) is the gravitational constant with respect to time interval \( t \), \( \alpha \) is a coefficient which decreases with time, \( C_I \) is the current iteration, \( M_I \) is the maximum number of iterations, and \( G(t_0) \) indicates the initial value of the gravitational constant.

The gravitational constant, \( G \) has a special role to play in the proper navigation of search space. It is because \( G \)'s value exponentially decreases slowly in the initial iterations which help in the exploration of the search space (Rather et al., 2019a, 2019b, 2019c, 2019d). Further, in the last iterations, \( G \)'s value decreases with a fast exponential pace and hence, aids in the convergence of optimal candidate solutions towards heavy agents. In 2017, Mijalili et al. have introduced chaotic normalization in GSA by incorporating chaotic sequences in gravitational constant for increasing convergence rate and removing stagnation in the local minima problem of GSA. The normalization function, \( C_i^{norm} \) with respect to interval \( t \) is given by Equation (3).
\[ C_i^{\text{norm}}(t) = \frac{(C_i(t) - a) \cdot (d - c)}{(b - a)} + c \]  
\[ (3) \]

Where \((a, b)\) are chaotic range values, \(i\) is the index of the chaotic map, and \((c, d)\) are normalized intervals. Moreover, \(c\) has a value of zero and \(d\) is given by Equation (4).

\[ d = MI - \frac{CI}{MI} \text{(Max-Min)} \]
\[ (4) \]

In Equation (4), \(\text{Max and Min}\) are adaptive intervals having a value of 20 and 1e-10, respectively. The final value of the gravitational constant is the addition of Equation (2) and (3) as shown in Equation (5).

\[ G^e(t) = C_i^{\text{norm}}(t) + G(t_0) e^{-\frac{CI}{MI}} \]
\[ (5) \]

According to Equation (5), \(G^e\) has both adaptive learning capability and chaotic ergodicity. On the theoretical side, the chaotic maps have the advantages of producing abrupt changes in the value of \(G\) which allows candidate solutions to get away from local minima traps.

Moreover, the calculation of total force exerted on the system due to masses can be calculated using Equation (6).

\[ F_i^d(t) = \sum_{j \neq i}^{m} \gamma_j F_j \]
\[ (6) \]

Whereas \(\gamma_j\) belongs to \([0, 1]\).

According to force law of motion, “The acceleration \(a\) of the agent is directly proportional to the force \(F\) applied by the agent and inversely proportional to the mass \(M\) of the agent”. It is calculated using Equation (7).

\[ a_i^d(t) = \frac{F}{M} \]
\[ (7) \]

To find the global optimum, it is important to calculate the position and velocity of the heavy mass. These can be represented mathematically as in Equation (8) and Equation (9), respectively.

\[ v_i^d(t + 1) = \gamma_j v_i^d(t) + a_i^d(t) \]
\[ (8) \]

\[ x_i^d(t + 1) = x_i^d(t) + a_i^d(t + 1) \]
\[ (9) \]

The pseudo-code and flowchart of CGSA are shown in Algorithm 1 and Figure 2, respectively. The optimization algorithms are comfortable in solving continuous benchmark test functions because it is simple for agents to explore the search space and find feasible candidate solutions.
However, the real test of intelligent techniques is solving complex non-linear test functions like engineering benchmarks where algorithms have to deal with constraints and inequalities.

5.1. Computational Complexity of CGSA

The optimization potential of HA is inversely proportional to its computational complexity. In other words, when a HA solves a real-world complex problem in less time, it depicts its time efficiency and practical applicability (Bansal (2018)). So, determining the time complexity of a HA is essential as far as its problem-solving potential is concerned. Therefore, the time complexity of CGSA has been calculated using big-O notation by considering the pseudo-code as depicted in Algorithm 1. It is important to mention that the initialization parameters like the number of variables, searcher agents, and stopping criterion have a significant contribution to the computational complexity of CGSA. So, the worst-case time complexity of CGSA is as follows:

\[ O(\text{CGSA}) = O(T(ND + N)) \]

Where, \( T \) = maximum number of iterations
\( N \) = population size
\( D \) = dimension of the benchmark function

In the next section, the CGSA has been applied to six mechanical and civil engineering design problems. It will be quite interesting to see how chaotic maps navigate the unknown search spaces of engineering design problems.

6. CGSA FOR ENGINEERING DESIGN PROBLEMS

The main aim of the optimization algorithm is to minimize the values of design parameters and the overall cost of the engineering design problem. The CGSA has been applied to six famous mechanical and civil engineering design frameworks namely Speed Reducer Design (SRD), Multiple Disc Clutch...
Brake Design (MDCBD), Three Bar Truss Design (TBTD), Stepped Cantilever Beam Design (SCBD), Gear Train Design (GTD), and Hydrodynamic Thrust Bearing Design (HTBD). The engineering problems consist of both equality and inequality constraints. To deal with the constraints, the penalty function method (PFM) has been employed. In PFM, constraints are converted into unconstrained...
functions so that HA can be used to solve them. The PFM can be mathematically represented in Equation (10).

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

(10)

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

In addition, CGSA has been compared with eleven stochastic techniques including CPSOGSA, Standard GSA, PSO, BBO, GA, DE, ACO, SCA, SSA, GWO, and PSOGSA for parameter value optimization and minimum cost of the design. For a fair comparison, all the algorithms were initialized with the same number of searcher agents and penalty constant value.

The experiment is conducted on a system that has a 64-bit operating system, 2.20 GHz Intel @ core i5 processor, 1TB hard disk, and 4GB RAM. Further, the implementation of the algorithms and engineering design problems has been done using MATLAB R2013a. The source codes are publicly available on Github i.e. https://github.com/SajadAHMAD1.

The experimental results of all the HAs were taken on a population size of 50 and maximum iterations of 500. Also, algorithms were repeated over 20 independent runs. The mean (or, average),

| Algorithm | Name of the Parameter | Value of the Parameter |
|-----------|-----------------------|------------------------|
| GSA       | Elitist Check (Number of Fittest Agents after Stopping Criterion) | 1 |
|           | Rpower (Power of Distance between Agents) | 1 |
|           | Min_flag (1: Minimum; 0: Maximum) | 1 |
| ACO       | Pheromone Update Constant | 1 |
|           | Initial Pheromone | 10 |
|           | Pheromone Sensitivity | 0.3 |
|           | Visibility Sensitivity | 0.1 |
| GA        | \(P_c\) (Crossover Probability) | 0.95 |
|           | \(P_m\) (Mutation Probability) | 0.001 |
|           | Er (Elitism) | 0.2 |
| DE        | Lower Bound of Scaling factor | 0.2 |
|           | Upper Bound of Scaling factor | 0.8 |
|           | PCR (Crossover Probability) | 0.8 |
| PSO       | C1, C2 (Personal and Social Constants) | 2 |
|           | Wmax (Maximum Inertia Weight) | 0.9 |
|           | Wmin (Minimum Inertia Weight) | 0.2 |
| BBO       | nKeep (Number of Habitats Retained after every Generation) | 0.2 |
|           | Pmutation (Mutation Probability) | 0.9 |
| SCA       | a (Constant) | 2 |
| SSA       | \(c_2, c_3\) (Random Numbers) | [0.1] |
| GWO       | \(c_2, c_3\) (Random Numbers) | [0.1] |
|           | a (Constant Vector) | Decrease from 2 to 0 |
| CPSOGSA   | \(\varphi_1, \varphi_2\) (Control Parameters) | 2.05 |
| PSOGSA    | Coefficient (k) | 23 |
|           | G(b) | 1 |
| CGSA      | Chaotic Initial Value | 20 |
standard deviation (STD), and median of 20 independent trails of the best solutions are reported. Further, CGSA1 to CGSA10 represents chaotic maps in the same ascending order as shown in Table 2.

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 pair-wise non-parametric signed Wilcoxon rank-sum test has been performed at a 5% significance level to statistically validate the simulation results. The reason behind selecting a Wilcoxon rank-sum test is that it uses median as a statistical measure which is better than average and STD. Moreover, in the Wilcoxon rank-sum test, the distribution of dataset is not considered. The null hypothesis consists of an algorithm having p-values less than 0.05 represented in experimental tables as NA (Not Applicable) because it can’t be compared with itself whereas alternate hypothesis include algorithm(s) with p-values greater than 0.05. Besides, a p-value of 1 shows the statistical equivalency of competing algorithms. Generally when the particular algorithm has minimum mean and standard deviation value; it is not compared with itself and represented in tables by N/A i.e. ‘not applicable’. Moreover, the chaotic maps other than the best one are represented by ‘NN’, that is, ‘Not Needed’. Also, the best cost values are highlighted in a bold font, and p values greater than 0.05 are underlined to indicate the competitiveness of the HA with the best one.

Meanwhile, it is important to initialize the parameters of the heuristic algorithms before using them in the optimization process. So, Table 3 shows the values of the initialization parameters of the participating algorithms. The next sections of the paper deal with the statistical and simulation analysis of the engineering design frameworks employing 11 different HAs including ten versions of CGSA.

6.1. Speed Reducer Design (SRD) Problem

The design of the SRD framework is illustrated in Figure 3. This engineering design problem is considered as one of the toughest benchmarks due to its stringent constraints and complicated search space. It has been seen that most of the exact and heuristic techniques show sub-optimal values for SRD. Minimization of the weight of the gearbox is the cost function of the SRD problem while considering 11 different constraints (Basset et al., 2019). Furthermore, it consists of numerical value restrictions on the bending stress, shaft transverse deflections, surface stress, and shaft stresses.

There are seven variables involved which are, width of the face ($x_1$), teeth module ($x_2$), teeth number ($x_3$), first shaft length ($x_4$), second shaft length ($x_5$), 1st shaft ($x_6$), and 2nd shaft ($x_7$) diameter, respectively. Moreover, design variables are represented by $l_1, l_2, l_3, l_4, l_5, l_6,$ and $l_7$, respectively in cost function and constraint equations. The mathematical formulation of the SRD problem is given as:

Consider $\vec{l} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [l_1, l_2, l_3, l_4, l_5, l_6, l_7]$

Minimize $f(\vec{l}) = 0.7854 \cdot l_2^2 \cdot (3.3333 \cdot l_3^2 + 14.9334 \cdot l_3 - 43.0934) - 1.508 \cdot (l_6^2 + l_7^2) + 7.4777 \cdot (l_6^3 + l_7^3) + 0.7854 \cdot (l_6^2 + l_7^2)$

Subject to

$s_1(\vec{l}) = \frac{27}{l_2^2 l_3^2} \leq 0,$

$s_2(\vec{l}) = \frac{397.5}{l_2^2 l_3^2} \leq 0,$
Figure 3. Speed Reducer Design Problem (Basset et al., 2019)

\[
\begin{align*}
  s_3 (\bar{I}) &= \frac{1.93 l_3^3}{l_4^3 l_6^3} \leq 0, \\
  s_4 (\bar{I}) &= \frac{1.93 l_3^3}{l_4^3 l_6^3} \leq 0, \\
  s_5 (\bar{I}) &= \frac{1.0}{110 l_6^3} \sqrt{\frac{745.0 l_4}{l_1}}^2 + 16.9 \times 10^6 - 1 \leq 0, \\
  s_6 (\bar{I}) &= \frac{1.0}{85 l_7^3} \sqrt{\frac{745.0 l_6}{l_3}}^2 + 1579 \times 10^6 - 1 \leq 0, \\
  s_7 (\bar{I}) &= \frac{l_2}{40} - 1 \leq 0, \\
  s_8 (\bar{I}) &= \frac{5 l_4}{l_1} - 1 \leq 0, \\
  s_9 (\bar{I}) &= \frac{1}{12 l_6} - 1 \leq 0, \\
  s_{10} (\bar{I}) &= \frac{1.5 l_6 + 1.9}{l_4} - 1 \leq 0, \\
  s_{11} (\bar{I}) &= \frac{1.1 l_4 + 1.9}{l_5} - 1 \leq 0,
\end{align*}
\]
Decision variable interval values:

\[2.6 \leq l_1 \leq 3.6,\]
\[0.7 \leq l_2 \leq 0.8,\]
\[17 \leq l_3 \leq 28,\]
\[7.3 \leq l_4 \leq 8.3,\]
\[7.8 \leq l_5 \leq 8.3,\]
\[2.9 \leq l_6 \leq 3.9,\]
\[5.0 \leq l_7 \leq 5.5,\]

### 6.1.1. SRD Simulation Results and Discussion

Tables 4a and 4b provide the comparative analysis of CGSA with 11 other HAs for the SRD problem. It is clear from Table 4b that all chaotic maps show the same cost function values. Moreover, ACO depicts superior performance than other algorithms as it has the minimum values for mean and standard deviation. The Wilcoxon rank-sum test also indicates the statistical superiority of the ACO. It is also evident from Table 4a that most of the algorithms give the same values for the design variables of the SRD problem.

Moreover, Figure 4 and Figure 5 depict the convergence curves of CGSA and other participating algorithms. It is clearly evident that ACO has superior performance at both 100 and 500 iterations. Besides, all CGSA versions gave the same values for the design cost function and the mean. The convergence curves of CGSA 1 to 10 overlap with CPSOGSA and GSA because of approximate mean and STD (Standard Deviation) values. This behavior is due to the exploitation capability of gravitational constant as its value decreases with the successive iterations that help agents to get attracted to the single heavy mass.

Furthermore, box plots are represented in Figure 6. It conveys that PSO, BBO, ACO, and DE have the best values nearby among each other. On the other hand, CPSOGSA, PSOGSA, SSA, SCA, GWO, GSA, and CGSA have also similar minimum values for the cost function because they have good exploration capabilities helping them to remain away from stagnation in local minima.

### 6.2. Gear Train Design (GTD) Problem

The configuration of the GTD problem is illustrated in Figure 7. The minimization of the gear ratio is the objective function. In the GTD problem, the angular velocity of the shafts is taken into consideration. It consists of four variables \(T_a, T_b, T_d,\) and \(T_f\). Basically, design variables (A, B, C, and D) are teeth numbers of the four gears \(T_a, T_b, T_d,\) and \(T_f\) represented by \(l_1, l_2, l_3,\) and \(l_4\), respectively. The GTD problem is mathematically formulated as:

Consider \(\vec{l} = [l_1, l_2, l_3, l_4] = [A, B, C, D],\)

Minimize \(f(\vec{l}) = \left(\frac{1}{6.931 - \frac{\text{floor}(l_1)\text{floor}(l_2)}{\text{floor}(l_3)\text{floor}(l_4)}}\right)^2\)

Subject to

\[12 \leq l_1 \leq 60,\]
Table 4a. Experimental results of SRD problem

| Algorithm  | $l_1$   | $l_2$   | $l_3$   | $l_4$   | $l_5$   | $l_6$   | $l_7$   |
|------------|---------|---------|---------|---------|---------|---------|---------|
| GSA        | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| PSO        | 5.5000  | 5.5000  | 5       | 5.5000  | 5       | 5       | 5.5000  |
| PSOGSA     | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CPSOGSA    | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| BBO        | 5.5     | 5.5     | 5       | 5.5     | 5       | 5       | 5.5     |
| GA         | 2.8573  | 0.7075  | 27.8294 | 8.1078  | 8.2770  | 2.9546  | 5.0055  |
| DE         | 5.5     | 5.5     | 5       | 5.5     | 5       | 5       | 5.5     |
| ACO        | 5       | 3       | 2       | 4       | 1       | 6       | 7       |
| SSA        | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| SCA        | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| GWO        | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA1      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA2      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA3      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA4      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA5      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA6      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA7      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA8      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA9      | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |
| CGSA10     | 3.6000  | 0.8000  | 17      | 7.3000  | 7.8000  | 3.9000  | 5.5000  |

Table 4b. Statistical results of SRD problem

| Algorithm  | Best   | Worst  | Mean   | STD    | Median | P values |
|------------|--------|--------|--------|--------|--------|----------|
| GSA        | 2.96e15 | 2.96e15 | 2.96e15 | 0.5130 | 2.96e15 | 0.000007 |
| PSO        | 3.56e11 | 3.56e11 | 3.56e11 | 1.25e-04 | 3.56e11 | 0.000007 |
| PSOGSA     | 2.96e15 | 2.96e15 | 2.96e15 | 0.5120 | 2.96e15 | 0.000007 |
| CPSOGSA    | 2.96e15 | 2.96e15 | 2.96e15 | 0.5130 | 2.96e15 | 0.000007 |
| BBO        | 3.56e11 | 3.56e11 | 3.56e11 | 1.25e-04 | 3.56e11 | 0.000007 |
| GA         | 6.25e16 | 8.01e16 | 7.39e16 | 4.31e15 | 7.44e16 | 0.000088 |
| DE         | 3.56e11 | 3.56e11 | 3.56e11 | 1.25e-04 | 3.56e11 | 0.000007 |
| ACO        | 7.29e10 | 7.29e10 | 7.29e10 | 1.56e-05 | 7.29e10 | N/A      |
| SSA        | 2.96e15 | 2.96e15 | 2.96e15 | 0.5120 | 2.96e15 | 0.000007 |
| SCA        | 2.96e15 | 2.96e15 | 2.96e15 | 0.5120 | 2.96e15 | 0.000007 |
| GWO        | 2.96e15 | 2.96e15 | 2.96e15 | 0.5120 | 2.96e15 | 0.000007 |
| CGSA1      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | 0.000007 |
| CGSA2      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA3      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA4      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA5      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA6      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA7      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA8      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA9      | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
| CGSA10     | 2.96e15 | 2.96e15 | 2.96e15 | 0.51299 | 2.96e15 | NN       |
12 \leq l_2 \leq 60,
12 \leq l_3 \leq 60,
12 \leq l_4 \leq 60,

6.2.1. GTD Simulation Results and Discussion

The GTD problem has been solved by using different versions of CGSA and eleven other HAs as shown in Table 5. All chaotic maps show the same results for the cost function of the GTD mechanical
framework with some little variation in the parameter values. Further, ACO shows optimal performance for the minimization of the design cost for the GTD problem as it provides minimum values for statistical measures. Besides, there is a huge gap in the best values of ACO and other algorithms. It is because ACO is highly suitable for combinatorial optimization problems.

Furthermore Figure 8 and Figure 9 graphically depict the comparative performance of the participating algorithms. It can be clearly seen that GSA has less convergence speed as it takes more computational time while finding the feasible regions of the solution space. Moreover, the convergence curve of ACO is linear that shows the same fitness values at both 100 and 500 iterations. Besides, the convergence curves of other algorithms overlap with each other indicating the same exploitation rate in selecting optimal candidate solutions for finding global minima.
The box plots of the GTD problem are shown in Figure 10. It depicts the optimal performance of ACO as it has small values for the median, lower and upper quartiles. Besides, the best values of other algorithms are approximately close to each other. Hence, they lie linearly at the upper part of the plot.

### 6.3. Three Bar Truss Design (TBTD) Problem

The schematic diagram of the TBTD (Basset et al., 2019) is provided in Figure 11. It is a special case of fractional programming as highly non-linear constraints are involved in the problem. The cost function of the TBTD framework is to minimize the volume of the truss. The design variables are

| Algorithm | $T_a$ | $T_b$ | $T_d$ | $T_f$ | Best | Worst | Mean | STD | Median | P values |
|-----------|-------|-------|-------|-------|------|-------|------|-----|--------|----------|
| GSA       | 12    | 12    | 60    | 12.3654 | 821.15 | 821.15 | 821.15 | 2.3e-13 | 821.15 | 0 |
| PSO       | 12    | 12    | 60    | 12.3241 | 821.15 | 821.15 | 821.15 | 2.32-13 | 821.15 | 0 |
| PSOGSA    | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.3e-13 | 821.15 | 0 |
| CPSOGSA   | 12    | 12    | 60    | 12.4713 | 821.15 | 821.15 | 821.15 | 2.3e-13 | 821.15 | 0 |
| BBO       | 12.72 | 12.53 | 60    | 12.9607 | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | 0 |
| GA        | 57.01 | 59.28 | 12.97 | 56.0551 | 1.46e08 | 2.7339 | 2.14e08 | 3.98e07 | 2.08e08 | 0 |
| DE        | 12    | 12    | 60    | 12.2905 | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | 0 |
| ACO       | 3     | 1     | 4     | 2     | 1.838 | 1.838 | 1.838 | 0 | 1.838 | N/A |
| SSA       | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.3e-13 | 821.15 | 0 |
| SCA       | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.3e-13 | 821.15 | 0 |
| GWO       | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | 0 |
| CGSA1     | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | 0 |
| CGSA2     | 12    | 12    | 60    | 12.4701 | 821.15 | 821.15 | 821.15 | 2.3e-13 | 821.15 | 0 |
| CGSA3     | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |
| CGSA4     | 12    | 12    | 60    | 12.6362 | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |
| CGSA5     | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |
| CGSA6     | 12.2079 | 12.7034 | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |
| CGSA7     | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |
| CGSA8     | 12    | 12    | 60    | 12.5447 | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |
| CGSA9     | 12    | 12    | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |
| CGSA10    | 12.6500 | 12.5176 | 60    | 12    | 821.15 | 821.15 | 821.15 | 2.33e-13 | 821.15 | NN |

Figure 8. Convergence curve of GTD problem (Iteration=100)
three cross-sectional areas \((A_1, A_2, A_3)\) represented mathematically by \(l_1\) and \(l_2\), respectively in the objective function. Moreover, \(P\) is the load subjected to the truss and \(D\) is the symmetrical length as shown in Figure 11.

The TBTD is mathematically represented as given by:

\[
\tilde{\mathbf{t}} = [\mathbf{l}_1, \mathbf{l}_2]
\]
Minimize \( f(\vec{l}) = (2\sqrt{2}l_1 + l_2) \times D \)

Subject to

\[ s_1(\vec{l}) = \frac{\sqrt{2}l_1 + l_2}{\sqrt{2}l_1^2 + 2l_1 l_2} p - \sigma \leq 0, \]
\[ s_2(\vec{l}) = \frac{\sqrt{2}l_1^2 + 2l_1 l_2}{\sqrt{2}l_2^2 + 2l_2 l_1} p - \sigma \leq 0, \]
\[ s_3(\vec{l}) = \frac{\sqrt{2}l_2^2 + 2l_2 l_1}{\sqrt{2}l_1^2 + 2l_1 l_2} p - \sigma \leq 0, \]

Variable interval value range:

\[ 0 \leq l_1 \leq 1, \]
\[ 0 \leq l_2 \leq 1. \]

Whereas \( D = 100, p = 2 \text{ KN per cm sq.}, \) and \( \sigma = 2 \text{ KN per cm sq} \)

6.3.1. **TBTD Simulation Results and Discussion**

Table 6 shows that PSO provides optimal performance for the minimization of the truss volume. Further, if we analyze the mean, STD, and best values, it can be seen that DE has similar statistical
results as that of PSO. This interpretation of results is amazingly validated by Wilcoxon rank-sum test as the p-value of DE is 1 that is greater than 0.05 indicating DE has the same performance as PSOGSA. In simpler terms, simulation results of PSO are not statistically significant as compared to PSO. Besides, GA has minimum values for the design parameters and ACO shows sub-optimal performance than other HAs. Also, CGSA 10 (tent map) provides optimal best values for the TBTD framework as compared to other chaotic maps.

Moreover, Figures 12 and 13 depict that DE and PSO have the same shape of convergence curves. On the other hand, the convergence behavior of CGSA is exponentially decreasing showing the high speed of exploitation, that is, enhanced capability in selecting optimal candidate solutions. Basically, ergodic sequences in tent map assist CGSA in taking agents from local minima solutions to optimal search regions which in turn helps in finding the best candidate solutions in less iterations. In addition,

Table 6. Experimental results of TBTD problem

| Algorithm | $l_1$ | $l_2$ | Best   | Worst  | Mean     | STD     | Median | P values |
|-----------|-------|-------|--------|--------|----------|---------|--------|----------|
| GSA       | 0.8299| 0.2794| 188.1522| 200    | 193.9115 | 3.7944  | 194.3463| 0.000088 |
| PSO       | 0.7868| 0.2880| 186.3859| 186.3859| 186.3859 | 5.79e-14| 186.3859| N/A      |
| PSOGSA    | 0.91  | 0.16  | 186.6257| 200    | 191.9061 | 3.8318  | 191.7462| 8.857e-05|
| CPSOGSA   | 0.7898| 0.4022| 186.9465| 200    | 192.6104 | 4.6741  | 190.8126| 0.000088 |
| BBO       | 0.7879| 0.2859| 186.3859| 186.497 | 186.3955 | 0.026692| 186.3866| 0.000088 |
| GA        | 0.0039| 0.0981| 1.521e07| 1.74e13| 8.819e11 | 3.896e12| 5.105e08| 0.000088 |
| DE        | 0.7868| 0.2880| 186.3859| 186.3859| 186.3859 | 4.259e-06| 186.3859| N/A      |
| ACO       | 1     | 2     | 400    | 400    | 400      | 0       | 400    | 0.000007 |
| SSA       | 0.79  | 0.29  | 186.3859| 186.3859| 186.3859 | 1.6585e-13| 186.3859| 8.857e-05|
| SCA       | 0.79  | 0.28  | 186.3859| 199.8759| 187.0736 | 3.0121  | 186.4037| 0.001324 |
| GWO       | 0.79  | 0.29  | 186.3859| 186.386 | 186.3859 | 1.6599e-05| 186.3859| 8.857e-05|
| CGSA1     | 0.9062| 0.1793| 188.4124| 200    | 194.848  | 3.6316  | 195.413 | NN       |
| CGSA2     | 0.9124| 0.1013| 186.6115| 200    | 194.7517 | 4.6195  | 194.9984| NN       |
| CGSA3     | 0.8077| 0.2997| 186.0696| 200    | 194.9312 | 3.4621  | 195.4647| NN       |
| CGSA4     | 0.8601| 0.1746| 187.1107| 200    | 192.5865 | 4.4705  | 191.1866| NN       |
| CGSA5     | 0.8627| 0.1629| 188.5133| 200    | 193.7242 | 4.3553  | 191.6802| NN       |
| CGSA6     | 0.8203| 0.2505| 186.624  | 200    | 192.5056 | 4.6457  | 191.4748| NN       |
| CGSA7     | 0.7919| 0.3484| 186.5018| 200    | 193.2682 | 3.861   | 193.0985| NN       |
| CGSA8     | 0.7948| 0.2956| 186.4499| 200    | 193.8439 | 4.6218  | 194.3577| NN       |
| CGSA9     | 0.7553| 0.3686| 187.1557| 200    | 193.5346 | 5.2619  | 192.3263| NN       |
| CGSA10    | 0.7553| 0.3415| 187.5237| 200    | 192.3073 | 3.6985  | 192.3073| 0.000087 |

Figure 12. Convergence curve of TBTD problem (Iteration=100)
SSA, SCA, PSOGSA, GWO, and CPSOGSA curves depict identical stochastic behavior indicating symmetrical values for the objective function. Besides, the box plots of the TBTD problem are shown in Figure 14. It depicts the sub-optimal performance of ACO as it has large cost function values.

6.4. Stepped Cantilever Beam Design (SCBD) Problem

The design of the five stepped cantilever beam problem (Thander and Vanderplaates, 1995) is shown in Figure 15. It has a rectangular shape and consists of ten design variables in which five are related to width ($X_1$ to $X_5$) and the other five are associated with the height ($h$) ($X_6$ to $X_{10}$) of the
5-stepped cantilever. The cost function of the SCBD problem is to minimize the volume of the beam subjected to load $P$.

The mathematical representation of the SCBD problem is given by:

Consider $\bar{X} = [X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}]$

Minimize $f(\bar{X}) = \sum_{i=1}^{5} X_i X_{i+5} l_i$

Where $[l_1, l_2, l_3, l_4, l_5] = 100\text{cm}$

Subject to

$s_1(\bar{X}) = \frac{600P}{X_1 X_2^2} - 14000 \leq 0,$

$s_2(\bar{X}) = \frac{X_1 X_2}{6P (l_5 + l_4)} - 14000 \leq 0,$

$s_3(\bar{X}) = \frac{6P (l_5 + l_4 + l_3)}{X_3 X_4^2} - 14000 \leq 0,$

$s_4(\bar{X}) = \frac{6P (l_5 + l_4 + l_3 + l_2 + l_1)}{X_5 X_6^2} - 14000 \leq 0,$

$s_5(\bar{X}) = \frac{P l_3^3}{3E} \left( \frac{1}{l_5} + \frac{7}{l_4} + \frac{19}{l_3} + \frac{37}{l_2} + \frac{61}{l_1} \right) - 2.7 \leq 0,$

$s_6(\bar{X}) = \frac{X_{10}}{X_5} - 20 \leq 0,$

$s_7(\bar{X}) = \frac{X_9}{X_4} - 20 \leq 0,$

$s_8(\bar{X}) = \frac{X_8}{X_3} - 20 \leq 0,$

Figure 15. Stepped Cantilever Beam Design Problem (Thander and Vanderplaats, 1995)
\[ s_{10}(\vec{X}) = \frac{X_7}{X_2} - 20 \leq 0, \]
\[ s_{11}(\vec{X}) = \frac{X_6}{X_1} - 20 \leq 0, \]

Decision variable interval values:

1 \leq X_1 \leq 5,
1 \leq X_2 \leq 5,
1 \leq X_3 \leq 5,
1 \leq X_4 \leq 5,
1 \leq X_5 \leq 5,
30 \leq X_6 \leq 65,
30 \leq X_7 \leq 65,
30 \leq X_8 \leq 65,
30 \leq X_9 \leq 65,
30 \leq X_{10} \leq 65,

6.4.1. SCBD Simulation Results and Discussion

The computational results of the SCBD problem are reported in Table 7a and 7b. It shows that CGSA (i.e. CGSA6) provides optimal performance for the minimization of the overall volume. Further, if we analyze the mean, STD, and best values, it can be seen that CPSOGSA and GSA have also similar statistical results. The Wilcoxon statistical test also validates this interpretation. Table 7b shows that the p-value of PSOGSA, CPSOGSA, SCA, SSA, GWO, and GSA is 1, which is greater than 0.05 specifying that all aforementioned algorithms have the same performance as that of CGSA. In simpler terms, the statistical results of CGSA are not statistically significant as compared to PSOGSA, CPSOGSA, SCA, SSA, GWO, and GSA. Besides, CGSA also gives optimal values to most of the design variables. Furthermore, the piecewise map (CGSA6) provides optimal performance as compared to other chaotic maps. It is because the piecewise map has intense randomized sequences on both halves of the y coordinate depicting high non-linear behavior. It helps CGSA to explore optimally complex search space without getting stuck in local minima while exploiting optimal regions. Besides, random chaotic behavior takes agents from local minima traps instantly towards optimal regions. This ergodic behavior helps CGSA to get optimal values for design parameters as compared to other algorithms.

The convergence curves of CGSA and 11 other competing algorithms are shown in Figures 16 and 17. It can be noticed that most of the algorithms have linear convergence behavior depicting stable intensification capability. However, when graphs are observed keenly at the initial iterations, it is seen that CGSA, BBO, and GWO depict fast convergence towards the global optimum. Moreover, CGSA also has better exploitation and solution quality than CPSOGSA, GSA, PSOGSA, ACO, PSO, and so on.

It is clearly evident that ACO and GA have large values for standard deviation showing poor exploitation power. Besides, they exhibit spikes in the box plot diagram as can be seen in Figure 18.
The whiskers are also depicting the outliers in the data which are mainly seen in ACO data as it has large values for lower and upper quartiles.

### 6.5. Multiple Disc Clutch Brake Design (MDCBD) Problem

The configuration of the MDCBD problem is shown in Figure 19. The purpose of this problem is to reduce the mass of the disc (Rao et al., 2011; Savsani et al., 2016). There are five design variables

| Algorithm | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| GSA       | 3.2281| 3.3817| 3.8900| 3.1510| 3.3513| 65    | 61.5921| 59.3357| 36.8860| 37.4950|
| PSO       | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    | 30    |
| PSOGSA    | 4.2914| 3.8721| 3.1513| 2.2444| 5     | 65    | 61.0345| 64.7153| 33.5445| 35.5124|
| CPSOGSA   | 3.8820| 4.1803| 2.9332| 4.6151| 2.6759| 54.6896| 54.5957| 59.3491| 31.1374| 42.5565|
| BBO       | 31.3860| 30.3910| 30.3612| 30.6013| 30.4641| 30.6931| 30.1263| 30.3842| 30.3842| 3    |
| GA        | 1.2150| 1.4220| 1.0546| 1.3613| 1.2769| 32.9029| 31.5331| 30.5242| 30.2772| 60.9335|
| DE        | 30    | 30    | 30.0558| 30    | 30    | 30    | 30    | 30    | 30    | 30    |
| ACO       | 7     | 7     | 9     | 10    | 3     | 4     | 8     | 1     | 6     | 5     |
| SSA       | 4.0316| 3.0016| 4.3395| 4.0432| 2.2811| 55.1687| 61.6312| 60.2411| 37.6815| 34.1967|
| SCA       | 4.5634| 4.3913| 2.9448| 5     | 2.6621| 53.3589| 47.9030| 52.3891| 34.7948| 30    |
| GWO       | 3.5906| 3.6769| 3.4673| 5     | 2.7668| 65    | 65    | 54.2787| 30    | 30    |
| CGSA1     | 3.2892| 3.5392| 3.0607| 2.9927| 2.9022| 63.6773| 54.6074| 53.2971| 43.5992| 44.4025|
| CGSA2     | 3.9775| 3.4339| 3.3043| 2.9449| 4.9305| 53.6472| 51.1540| 47.9914| 47.9398| 43.1204|
| CGSA3     | 3.3860| 4.8437| 3.0073| 2.8357| 3.4683| 62.0362| 46.3658| 48.2565| 47.0358| 48.4539|
| CGSA4     | 5     | 3.7360| 2.8108| 3.0360| 3.0949| 57.8483| 49.1089| 47.9339| 52.7594| 40.6368|
| CGSA5     | 4.5499| 2.8037| 3.5347| 3.8873| 2.6650| 51.2341| 57.2564| 43.9380| 56.1908| 54.5812|
| CGSA6     | 3.3252| 2.8024| 2.6617| 2.6684| 2.5724| 58.0999| 58.2515| 52.0070| 48.8929| 42.1294|
| CGSA7     | 3.6314| 3.1164| 3.0742| 2.4638| 2.1320| 57.7300| 54.6622| 48.9157| 48.3239| 49.4451|
| CGSA8     | 4.5453| 3.5932| 3.2413| 3.0363| 2.8824| 56.5762| 50.4123| 45.9204| 51.1599| 39.9100|
| CGSA9     | 3.9119| 3.4305| 3.2762| 2.2226| 3.4124| 59.0000| 52.5002| 50.4436| 46.1398| 51.3860|
| CGSA10    | 3.1925| 3.4392| 2.5475| 3.5662| 3.1410| 58.2093| 51.5501| 52.2613| 47.9114| 41.5285|

The whiskers are also depicting the outliers in the data which are mainly seen in ACO data as it has large values for lower and upper quartiles.

### Table 7a. Experimental results of SCBD problem

### Table 7b. Statistical results of SCBD problem
involved namely radius of the disc (inner $r_i$ and outer $r_o$), disc thickness ($t$), operating force ($F$), and the number of disc surfaces ($z$).

The mathematical formulation of MDCBD framework is given by:

Consider $\mathbf{X} = [r_i, r_o, t, F, z]$

Minimize $f(\mathbf{X}) = \pi t (r_o^2 - r_i^2)$

Subject to
\begin{align*}
s_1(X) &= r_o - r_i - \Delta r \geq 0, \\
s_2(X) &= l_{\text{max}} - (z + 1)(t + \delta) \geq 0, \\
s_3(X) &= p_{\text{max}} - p_{r2} \geq 0, \\
s_4(X) &= p_{\text{max}}P_{\text{arma}} - p_{r2}P_{s1} \geq 0, \\
s_5(X) &= P_{\text{arma}} - P_{s1} \geq 0, \\
s_6(X) &= T_{\text{max}} - V_{s1} \geq 0, \\
s_7(X) &= M_h - sM_s \geq 0, \\
s_8(X) &= T \geq 0,
\end{align*}

Such that

\begin{align*}
M_h &= \frac{2}{3}\mu Fz \frac{r_o^3 - r_i^3}{r_o^2 - r_i^2}, \\
P_{r2} &= \frac{F}{\pi \left( r_o^2 - r_i^2 \right)}, \\
P_{s1} &= \frac{I_z \pi h}{30 \left( M_h - M_f \right)}, \\
V_{s1} &= \frac{2\pi N \left( r_o^3 - r_i^3 \right)}{90 \left( r_o^2 - r_i^2 \right)}, \\
T &= \frac{I_z \pi h}{30 \left( M_h - M_f \right)}.
\end{align*}

Decision variable interval values:

\begin{align*}
60 \leq r_i \leq 80, \\
90 \leq r_o \leq 110, \\
105 \leq t \leq 3, \\
600 \leq F \leq 1000, \\
2 \leq z \leq 9,
\end{align*}

Figure 18. Box plots of SCBD problem
Also,
\[ z_{max} = 10, \quad v_{armax} = 10 \text{m s}^{-1}, \quad \mu = 0.5, s = 1.5, \]
\[ M_s = 40 \text{Nm}, \quad M_f = 3 \text{Nm}, \quad n = 250 \text{ rpm}, \quad \rho_{max} = 1 \text{Mpa}, \]
\[ I_z = 55 \text{kg mm sq.}, \quad T_{max} = 15 \text{s}, \quad F_{max} = 1000 \text{N}, \]
\[ r_{imin} = 9, \quad r_{omax} = 110 \text{m} \]

6.5.1. MDCBD Simulation Results and Discussion
The experimental results of the MDCBD problem are reported in Table 8. It can be seen that ACO provides the best values for cost function and minimum values for mean as compared to other algorithms. Further, all chaotic maps, PSOGSA, SCA, SSA, GWO, CGSAPSO, and GSA show the same values for design parameters, mean and STD.

The convergence behavior of the MDCBD engineering benchmark is reported in Figures 20 and 21. It can be comprehensibly seen that CGSA has fast convergence speed as compared to standard...
GSA. Moreover, the curves of SSA, SCA, GWO, PSOGSA, and CPSOGSA are interlinked and coupled with each other indicating identical and symmetrical convergence patterns. The convergence maps also convey that most of the HAs have the same values for the objective function and stable exploitation capability as far as the MDCBD problem is concerned.

Besides, Figure 22 depicts box plots for the MDCBD problem. The upper and lower quartiles of DE, PSO, GA, BBO, and ACO are at the same level showing similarity in statistical outcomes. Further, the GSA, PSOGSA, CPSOGSA, SSA, SCA, GWO, and CGSA also have objective values in the same numerical neighborhood.

Table 8. Experimental results of MDCBD problem

| Algorithm | $r_1$ | $r_2$ | $t$ | F | z | Best | Worst | Mean | STD | Median | P values |
|-----------|-------|-------|-----|---|---|------|-------|------|-----|--------|----------|
| GSA       | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| PSO       | 4.2949| 4.3451| 5.7329 | 7.2451 | 4.2330 | 10.0 e05 | 4.71 e07 | 3.41 e07 | 1.81 e07 | 4.71 e07 | 0.000007 |
| PSOGSA    | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CPSOGSA   | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| BBO       | 2.8180| 2.8830| 4.8479 | 8.9562 | 5.1895 | 4.71 e07 | 4.71 e07 | 4.71 e07 | 2.42 e03 | 4.71 e07 | 0.000008 |
| GA        | 73.5825| 109.4542 | 1.8002 | 934.7027 | 8.9387 | 1.47e16 | 1.76e16 | 1.63e16 | 8.69e14 | 1.65e16 | 0.000008 |
| DE        | 9     | 9     | 8.1573 | 8.7293 | 2.7618 | 10.0 e05 | 4.71 e07 | 1.19 e07 | 8.28 e06 | 10.0 e05 | N/A |
| ACO       | 1     | 2     | 5    | 4   | 3  | 3.15e08 | 3.15e08 | 3.15e08 | 1.22e-07 | 3.15e08 | 0.000007 |
| SSA       | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| SCA       | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| GWO       | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA1     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA2     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA3     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA4     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA5     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA6     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA7     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA8     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA9     | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |
| CGSA10    | 60    | 90    | 3   | 600 | 2  | 2.07e14 | 2.07e14 | 2.07e14 | 0.0641 | 2.07e14 | 0.000007 |

Figure 20. Convergence curve of MDCBD problem (Iteration=100)
6.6. Hydrodynamic Thrust Bearing Design (HTBD) Problem

The design of the HTBD problem is illustrated in Figure 23. HTBD is a classical mechanical engineering design problem that has the purpose of minimizing the power loss of the framework (Rao et al., 2011). It consists of four design variables namely radius (step radius (R) and recess radius ($R_0$)), viscosity ($\mu$), and rate of flow (Q).

The mathematical formulation of HTBD problem is given by:

Consider $\vec{X} = [R \ R_0 \ \mu \ Q]$

Minimize $f(\vec{X}) = \frac{QP_0}{0.7} + E_f$

Figure 21. Convergence curve of MDCBD problem (Iteration=500)

Figure 22. Box plots of MDCBD problem
Subject to
\[ s_1(\tilde{X}) = W - W_s \geq 0, \]
\[ s_2(\tilde{X}) = P_{\text{max}} - P_0 \geq 0, \]
\[ s_3(\tilde{X}) = \Delta T_{\text{max}} - \Delta T \geq 0, \]
\[ s_4(\tilde{X}) = h - h_{\text{min}} \geq 0, \]
\[ s_5(\tilde{X}) = R - R_0 \geq 0, \]
\[ s_6(\tilde{X}) = 0.001 - \frac{\gamma}{gP_0} \left( \frac{Q}{2\pi Rh} \right) \geq 0, \]
\[ s_7(\tilde{X}) = 5000 - \frac{W}{\pi \left( R^2 R_0^2 \right)} \geq 0, \]

Decision variable interval values:
Table 9. Experimental results of HTBD problem

| Algorithm | $R$  | $R_o$ | $\mu$ | $Q$  | Best | Worst | Mean | STD | Median | P values |
|-----------|------|-------|-------|------|------|-------|------|-----|--------|----------|
| GSA       | 2.6935 | 1.9866 | 6.89e06 | 1,7350 | 6.20e42 | 3.93e62 | 2.81e61 | 9.26e61 | 3.01e55 | 0.000088 |
| PSO       | 1     | 1     | 16    | 16   | 2.36e11 | 2.73e12 | 2.36e11 | 3.80e05 | 2.36e11 | 0.250000 |
| PSSGSA    | 1     | 1     | 1e6   | 16   | 2.36e11 | 2.73e12 | 2.61e11 | 5.99e12 | 2.36e11 | 0.392e-04 |
| CPSOGSA   | 1     | 1     | 1e6   | 16   | 2.36e11 | 2.73e12 | 4.80e11 | 7.53e12 | 2.36e11 | 0.000007 |
| BBO       | 1.8794 | 3.3882 | 4.3489 | 12.7023 | 1.57e12 | 2.70e12 | 2.25e12 | 3.22e11 | 2.33e12 | 0.000088 |
| GA        | 9.1017 | 14.8046 | 1.47e07 | 1.1259 | 6.45e90 | 2.54e93 | 6.32e92 | 5.97e92 | 5.18e92 | 0.000088 |
| DE        | 1     | 1     | 16    | 16   | 2.36e11 | 2.36e11 | 2.36e11 | 3.13e-05 | 2.36e11 | 0.000088 |
| ACO       | 1     | 3     | 4     | 2    | 7.35e13 | 7.35e13 | 7.35e13 | 0.0321 | 7.35e13 | N/A      |
| SSA       | 1     | 1     | 1e6   | 16   | 2.36e11 | 2.73e12 | 2.36e11 | 6.26e-05 | 2.36e11 | 7.74e-06 |
| SCA       | 1     | 1     | 1e6   | 16   | 2.36e11 | 2.73e12 | 2.36e11 | 6.26e-05 | 2.36e11 | 7.74e-06 |
| GWO       | 1     | 1     | 1e6   | 16   | 2.36e11 | 2.73e12 | 2.36e11 | 6.26e-05 | 2.36e11 | 7.74e-06 |
| CGSA1     | 2.2385 | 1.2126 | 5.17e06 | 12.7936 | 8.93e64 | 1.23e65 | 9.00e63 | 2.84e64 | 7.79e56 | 0.000007 |
| CGSA2     | 1.1677 | 1.6634 | 1.58e07 | 1.0139 | 6.39e44 | 1.29e66 | 7.29e64 | 2.89e65 | 1.93e58 | NN       |
| CGSA3     | 2.4297 | 2.7208 | 1.20e07 | 13.9508 | 3.92e44 | 1.65e64 | 8.60e62 | 3.68e63 | 1.87e59 | NN       |
| CGSA4     | 2.0440 | 1.6001 | 4.41e06 | 6.0922 | 1.67e45 | 3.43e62 | 1.97e61 | 7.65e61 | 1.50e56 | 0.000088 |
| CGSA5     | 6.0519 | 5.7068 | 7.58e06 | 11.2631 | 2.03e24 | 3.81e63 | 2.80e62 | 8.97e62 | 1.33e58 | NN       |
| CGSA6     | 1.3580 | 1.5969 | 1.25e06 | 11.0145 | 8.92e40 | 2.02e66 | 1.03e65 | 4.50e65 | 2.12e56 | NN       |
| CGSA7     | 8.5169 | 8.4366 | 9.18e06 | 4.55235 | 4.89e47 | 1.37e66 | 7.11e64 | 3.06e65 | 4.06e57 | NN       |
| CGSA8     | 2.5840 | 2.2464 | 7.20e06 | 2.6410 | 5.90e47 | 1.80e65 | 9.03e63 | 4.03e64 | 2.06e57 | NN       |
| CGSA9     | 5.7715 | 5.7724 | 1.50e07 | 5.6456 | 1.18e38 | 1.98e65 | 1.00e64 | 4.44e64 | 6.46e57 | NN       |
| CGSA10    | 2.0909 | 2.1203 | 4.53e06 | 3.5464 | 1.55e51 | 3.66e67 | 1.94e66 | 8.17e66 | 6.21e59 | NN       |

$1 \leq R \leq 16$,  
$1 \leq R_o \leq 16$,  
$1 \times 10^{-6} \leq \mu \leq 16 \times 10^{-6}$,  
$1 \leq Q \leq 16$
6.6.1. HTBD Simulation Results and Discussion

Table 9 provides statistical results of optimization algorithms for the HTBD problem. It is indicated that DE and PSO provide minimum results for the cost function. It is because both the algorithms have good exploitation properties that help agents to reach the global optimum in less iterations. Moreover, PSO has pbest and gbest parameters which are designed particularly for global and local search. Further, DE has mutation and selection operators which are not only robust in finding the diversity of candidate solutions but also selecting optimal candidate agents from the feasible solutions.

Similarly, Figures 24 and 25 depict convergence curves of simulation results taken from 20 independent runs of algorithms. It can be seen that curves of DE, PSO, and ACO are having proximity...
in objective values. However, CGSA and GSA take more computational time as compared to other HAs in finding the global optimum. Moreover, box plots of the algorithms are represented in Figure 26. It depicts that most of the algorithms have identical statistical values.

6.7. Overall Simulation Results Discussion

The experimental results indicate that chaotic maps were quite successful in showing optimal results for classical engineering design problems. It was amazing to see that CGSA always showing optimal results as compared to standard GSA. Besides, CGSA provides competitive results to all design problems and was always in the top 3 best performance algorithms. Besides, CGSA shows efficient results for SCBD, TBTD, and MDCBD problems. Moreover, on three design problems, all algorithms including CGSA have shown the same statistical outcomes namely GTD, SCBD, and MDCBD. Also, the tent, iterative, and piecewise chaotic maps have shown efficient results for TBTD, SCBD, and HTBD problems, respectively. All these maps have a high frequency of randomization in their chaotic behavior providing non-linearity which is essential for agent optimization. In simpler terms, the chaotic sequences abruptly change states while taking agents from local minima regions to optimal zones resulting in the high convergence towards the global minimum.

For comparative analysis, eleven algorithms were utilized in which ACO has shown optimal results to SRD and GTD frameworks. On the other hand, DE also performed well on MDCBD and HTBD design problems. To sum up, the obviously chaotic maps were efficient in increasing the overall performance of standard GSA. It is evident from the experimental results that standard GSA was always way behind the best results of CGSA. Also, CGSA has been successful in maintaining the adaptive capability of gravitational constant and also utilizing chaotic sequences for overcoming premature convergence and stagnation in local minima problems of standard GSA. Therefore, the use of chaotic maps in standard GSA is a new beginning for exciting research in different areas of GSA, particularly convergence behavior study, time complexity investigation, GSA variant comparative analysis, and solving practical application problems.

7. CONCLUSION AND FUTURE DIRECTIONS

In this paper, ten chaotic maps were combined with the gravitational constant of GSA and have been applied to six engineering design problems. The chaotic maps have shown efficient performance on most of the design problems especially stepped cantilever beam, three bar truss design, and hydrodynamic thrust bearing design frameworks. Further, they have shown competitive results on other problems too. From the ten chaotic maps, tent, iterative, and piecewise chaotic maps were efficient in showing best results in the minimization of the objective functions of design benchmarks. Generally speaking, CGSA has been quite successful in overcoming difficulties of standard GSA. Besides, chaotic sequences were superior as compared to random number generators as they were efficient in taking agents from sub-optimal regions of local minima which in turn helps in fast convergence of the CGSA algorithm.

For future studies, it will be quite interesting to utilize stochasticity, ergodicity, and complex non-linear motion properties of chaotic maps in intelligent techniques for overcoming the drawbacks of premature convergence, sensitive randomization, and entrapment in local minima. Moreover, CGSA can be applied to IIR filter design, economic dispatch, and MLP (Multi-Layer Perceptron) neural network training problems.

In addition, CGSA has a huge potential for hybridization with other state of the art algorithms such as PSO, DE, GA, and so on. Besides, for improving the performance of the CGSA, complex penalty functions can be utilized which will be strict against constraint violation and can accordingly penalize the cost function of the algorithms. Lastly, it will be amazing to combine other chaotic maps with GSA for studying their effects on convergence and the local exploitation rate of the standard GSA.
REFERENCES

Abed-alguni, B. H. (2019). Island-based Cuckoo Search with Highly Disruptive Polynomial Mutation. *International Journal of Artificial Intelligence, 17*(1), 57–82.

Abed-alguni, B. H., & Paul, D. J. (2018). Hybridizing the Cuckoo Search Algorithm with Different Mutation Operators for Numerical Optimization Problems. *Journal of Intelligent Systems, 0*(0), 1–20. doi:10.1515/jjisys-2018-0531

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.sofr.2007.09.063

Bansal, S. (2018). Nature-inspired-based multi-objective hybrid algorithms to find near-OGRs for optical WDM systems and their comparison. In *Handbook of Research on Biomimicry in Information Retrieval and Knowledge Management* (pp. 175–211). IGI Global. doi:10.4018/978-1-5225-3004-6.ch011

Bansal, S. (2019). A comparative study of nature-inspired metaheuristic algorithms in search of near-to-optimal Golomb rulers for the FWM crosstalk elimination in WDM systems. *Applied Artificial Intelligence, 33*(14), 1199–1265. doi:10.1080/08839514.2019.1683977

Bansal, S., Gupta, N., & Singh, A. K. (2017b). Nature–inspired metaheuristic algorithms to find near–OGR sequences for WDM channel allocation and their performance comparison. *Open Mathematics, 15*(1), 520–547. doi:10.1515/math-2017-0045

Bansal, S., & Sharma, K. (2018). Nature-inspired-based modified multi-objective BB-BC algorithm to find near-OGRs for optical WDM systems and its performance comparison. In *Handbook of Research on Biomimicry in Information Retrieval and Knowledge Management* (pp. 1–25). IGI Global. doi:10.4018/978-1-5225-3004-6.ch001

Bansal, S., Singh, A. K., & Gupta, N. (2017a). Optimal Golomb ruler sequences generation for optical WDM systems: a novel parallel hybrid multi-objective bat algorithm. *Journal of the Institution of Engineers (India): Series B, 98*(1), 43–64.

Baojiang, Z., & Shiyoung, L. (2007). Solving the neuro-fuzzy controller (NFC) using ACO. *Journal of Systems Engineering and Electronics, 18*(3), 603–601. doi:10.1016/S1004-4132(07)60135-2

Basset, M. A., Wang, G. G., Sangaiah, A. K., & Rushdy, E. (2019). Hybridization of KH and CS algorithms for solving Design Problems. *Multimedia Tools and Applications, 78*(4), 3861–3884. doi:10.1007/s11042-017-4803-x

Beigvand, S. D., Abdil, H., & La Scala, M. (2016). Combined heat and power economic dispatch problem using GSA. *Electric Power Systems Research, 133*, 160–172. doi:10.1016/j.epsr.2015.10.007

Beigvand, S. D., Abdil, H., & La Scala, M. (2017). A general model for energy hub economic dispatch. *Applied Energy, 190*, 1090–1111. doi:10.1016/j.apenergy.2016.12.126

Bhowmik, A. R., & Chakraborty, A. K. (2015). Solution of optimal power flow using non dominated sorting multi-objective opposition based GSA. *International Journal of Electrical Power & Energy Systems, 64*, 1237–1250. doi:10.1016/j.ijepes.2014.09.015

Cordon, O., Damas, S., & Santamari, J. (2006). A fast and accurate approach for 3D image registration using the scatter search evolutionary algorithm. *Pattern Recognition Letters, 27*(11), 1191–1200. doi:10.1016/j.patrec.2005.07.017

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
Du, Y., & Li, B. (2008). Dynamic optimization through PSO. *Information Sciences, 178*(15), 3096–3109. doi:10.1016/j.ins.2008.01.020

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., & Alavi, A. H. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation, 17*(12), 4831–4845. doi:10.1016/j.cnsns.2012.05.010

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. (2012). 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. (2012). 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

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

Javidi, M., & Hosseinipourfard, R. (2015). Chaos genetic algorithm instead of genetic algorithm. *The International Arab Journal of Information Technology, 12*(2), 163–168.

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

Kalinia, A., & Karabogab, N. (2005). Artificial immune algorithm for IIR filter design. *Engineering Applications of Artificial Intelligence, 18*(8), 919–929. doi:10.1016/j.engappai.2005.03.009

Karakuzu, C. (2008). Fuzzy controller training using particle swarm optimization for controlling complex systems. *ISA Transactions, 47*(2), 229–239. doi:10.1016/j.isatra.2007.09.003 PMID:17976603

Kellert, S. H. (2017). *In the wake of chaos: unpredictable order in dynamical systems*. University of Chicago Press.

Kennedy, J., & Eberhart, R. C. (1993). Particle Swarm optimization. *International Conference on Neural Networks, 1992-1948*. doi:10.1109/ICNN.1995.488968

Kim, T. H., Maruta, I., & Sugie, T. (2008). Employing PSO for PID controller tuning. *Automatica, 44*(4), 1104–1110. doi:10.1016/j.automatica.2007.08.017

Kumar, A., Misra, R. K., Singh, D., Mishra, S., & Das, S. (2019). The spherical search algorithm for bound-constrained global optimization problems. *Applied Soft Computing, 85*, 105734. doi:10.1016/j.asoc.2019.105734

Kwon, S., Kim, B. C., Hwang, H., Mun, D., & Han, S. (2016). Enhancement of equipment information sharing using three-dimensional computer-aided design simplification and digital catalog techniques in the plant industry. *Concurrent Engineering, 24*(3), 275–289. doi:10.1177/1063293X16638709

Li, S. (2016). Discrete chaotic gravitational search algorithm for unit commitment problem. *International Conference on Intelligent Computing Theories and Application, LNCS 9772*, 757-769. doi:10.1007/978-3-319-42294-7_67

Liu, Y., Yi, Z., Wu, H., Ye, M., & Chen, K. (2008). Clustering through Tabu search. *Information Sciences, 178*(12), 2680–2704. doi:10.1016/j.ins.2008.01.022

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

Miraljili, 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

Mittal, H. (2016). Chaotic Kbest gravitational search algorithm (CKGSA). *9th International Conference on Contemporary Computing (IC3).* doi: doi:10.1109/IC3.2016.788052

Mittal, H., & Saraswat, M. (2018). CKGSA based fuzzy clustering method for image segmentation of RGB-D images. *Eleventh International Conference on Contemporary Computing, 1-6*. doi:10.1109/IC3.2018.8530568
Mohamed, A. K., & Mohamed, A. W. (2019c). Real-parameter unconstrained optimization based on enhanced AGDE algorithm. In Machine Learning Paradigms: Theory and Application (pp. 431–450). Springer. doi:10.1007/978-3-030-02357-7_21

Mohamed, A. W., Hadi, A. A., & Jambi, K. M. (2019b). Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. Swarm and Evolutionary Computation, 50, 100455. doi:10.1016/j.swevo.2018.10.006

Mohamed, A. W., Hadi, A. A., & Mohamed, A. K. (2019a). Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. International Journal of Machine Learning and Cybernetics, 1–29. doi:10.1007/s13042-019-01053-x

Nezamabadi-pour, H., Saryazdi, S., & Rashedi, E. (2006). Edge detection using ant algorithms. Soft Computing, 10(7), 623–628. doi:10.1007/s00500-005-0511-y

Precup, R. E., & David, R. C. (2019). Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems. Butterworth-Heinemann, Elsevier.

Rao, R. V., Svani, S. J., & Vakharia, D. P. (2011). Teaching-Learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer Aided Design, 40(5-6), 303–315. doi:10.1016/j.cad.2010.12.015

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. 2019.8869279 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. In 2019 Elsevier International Conference on Advances in Electronics, Electrical, and Computational Intelligence (ICAEEC-2019). 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.

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., & Sharma, N. (2017). GSA-BBO Hybridization Algorithm. International Journal of Advance Research in Science and Engineering, 6, 596–608.

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

Saha, S.K., & Kar, R., Mandal, & Ghoshal, S.P. (2014). Gravitational Search Algorithm: Application to the IIR Filter Design. King Saud Journal of Engineering Sciences, 26(1), 69–81.

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

Savani, P., & Savani, V. (2016). Passing vehicle search (PVS): A novel metaheuristic algorithm. Applied Mathematical Modelling, 40(5-6), 3951–3978. doi:10.1016/j.apm.2015.10.040
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)*, 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

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

Talatahari, S., Azar, B. F., Sheikhholeslami, 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

Tan, X., & Bhanu, B. (2006). Application of GA in fingerprinting. *Pattern Recognition, 39*(3), 465–477. doi:10.1016/j.patcog.2005.09.005

Tang, K. S., Man, K. F., Kwong, S., & He, Q. (1999). Genetic algorithms and their applications. *IEEE Signal Processing Magazine, 13*(6), 22–37. doi:10.1109/79.543973

Thander, P. B., & Vanderplaats, G. N. (1995). Survey of discrete optimization for structural engineering. *Journal of Structural Engineering, 121*(2), 301–306. doi:10.1061/(ASCE)0733-9445(1995)121:2(301)

Wang, G. G., Deb, S., Gandomi, A. H., Zhang, Z., & Alavi, A. H. (2016). Chaotic cuckoo search. *Soft Computing, 20*(9), 3349–3362. doi:10.1007/s00500-015-1726-1 PMID:26880874

Wang, G. G., Guo, L., Gandomi, A. H., Hao, G.-S., & Wang, H. (2014). Chaotic krill herd algorithm. *Information Sciences, 274*, 17–34. doi:10.1016/j.ins.2014.02.123

Wang, M., Wan, Y., Ye, Z., Gao, X., & Lai, X. (2018). A band selection method for airborne hyperspectral image based on the chaotic binary gravitational search algorithm. *Neurocomputing, 273*, 57–67. doi:10.1016/j.neucom.2017.07.059

Wang, N., Liu, L., & Liu, L. (2001). Genetic algorithm in chaos. *OR Transactions, 5*, 1–10.

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

Yang, L. J., & Chen, T. L. (2002). Application of chaos in genetic algorithms. *Communications in Theoretical Physics, 38*(2), 168–172. doi:10.1088/0253-6102/38/2/168

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*(2), 82–1999. doi:10.1109/4235.771163

Zhenyu, G., Bo, C., Min, Y., & Binggang, C. (2006). Self-adaptive chaos differential evolution. *International Conference on Natural Computation: Advances in Natural Computation, 972-975.*

---

Sajad Ahmad Rather completed his M.Tech in Computer Science and Engineering from the SRM University, Chennai, India, in 2016. Currently, he is pursuing his doctorate in Artificial Intelligence and Machine Learning at Pondicherry University, India. He has publications in reputed international journals and presented his research in both national and international conferences (Springer, Elsevier, ACM, IEEE, etc.). He is a member of IEEE, ACM, ISTE, IEI, IETE, and IAENG. Moreover, he has also authored three books. He is also a reviewer of IGI Global and International Journal of Intelligent Computing and Cybernetics (IJICCC) Emerald. Besides, he is interested in Artificial Intelligence (AI), Swarm Intelligence, Machine learning (ML), Meta-heuristics, and Engineering Optimization.

P. Shanthi Bala is an Assistant Professor in the Department of Computer Science, School of Engineering and Technology, Pondicherry University, India. She is an expert in Distributed Computing and Ontology. She has published research in peer reviewed and SCI international journals. Moreover, she has chaired number of international conferences on Machine learning and Artificial Intelligence. Her research interests include Artificial Intelligence, Machine Learning, Neural Networks, Internet of Things (IoT), Deep Learning, Distributed Computing, and Ontology.