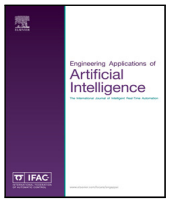
Engineering Applications of Artificial Intelligence 95 (2020) 103847

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Fuzzy mutation embedded hybrids of gravitational search and Particle Swarm Optimization methods for engineering design problems

Devroop Kar [a], Manosij Ghosh [a], Ritam Guha [a,*], Ram Sarkar [a,*], Laura Garcia-Hernandez [b], Ajith Abraham [c]

[a] Computer Science and Engineering Department, Jadavpur University, 188, Raja S.C. Mallick Road, Kolkata – 700032, West Bengal, India
[b] Area of Project Engineering, Universidad de Córdoba, Spain
[c] Scientific Network for Innovation and Research Excellence, Machine Intelligence Research Labs (MIR Labs) Auburn, WA 98071, USA

ARTICLE INFO

Keywords:
Fuzzy logic
Mutation
Gravitational search algorithm
Particle swarm optimization
Hybrid
Engineering design

ABSTRACT

Gravitational Search Algorithm (GSA) and Particle Swarm Optimization (PSO) are nature-inspired, swarm-based optimization algorithms respectively. Though they have been widely used for single-objective optimization since their inception, they suffer from premature convergence. Even though the hybrids of GSA and PSO perform much better, the problem remains. Hence, to solve this issue, we have proposed a fuzzy mutation model for two hybrid versions of PSO and GSA — Gravitational Particle Swarm (GPS) and PSOGSA. The developed algorithms are called Mutation based GPS (MGPS) and Mutation based PSOGSA (MPSOGSA). The mutation operator is based on a fuzzy model where the probability of mutation has been calculated based on the closeness of particle to population centroid and improvement in the particle value. We have evaluated these two new algorithms on 23 benchmark functions of three categories (unimodal, multimodal and multimodal with fixed dimension). The experimental outcome shows that our proposed model outperforms their corresponding ancestors, MGPS outperforms GPS 13 out of 23 times (56.52%) and MPSOGSA outperforms PSOGSA 17 times out of 23 (73.91%). We have also compared our results against those of some recently proposed optimization algorithms such as Sine Cosine Algorithm (SCA), Opposition-Based SCA, and Volleyball Premier League Algorithm (VPL). In addition, we have applied our proposed algorithms on some classic engineering design problems and the outcomes are satisfactory. The related codes of the proposed algorithms can be found in this link: Fuzzy-Mutation-Embedded-Hybrids-of-GSA-and-PSO.

1. Introduction

An optimization problem maximizes or minimizes a real function by systematically choosing input values from an allowed set. This problem is of particular interest in the fields of operations research and certain engineering applications. There are major subfields to this section of mathematics including convex programming, stochastic programming and optimization, meta-heuristic programming, etc. While meta-heuristics do not guarantee that the best solution will be found, they are widely used to find good approximate (optimal) solutions for many complicated optimization problems. Gravitational Search Algorithm (GSA) (Rashedi et al., 2009), Particle Swarm Algorithm (PSO) (Eberhart and Kennedy, 1995), Genetic Algorithm (Schott, 1995), Cuckoo search algorithm (Gandomi et al., 2013), Grey Wolf Algorithm (Dhargupta et al., 2020) are some of the effective meta-heuristic algorithms based on natural phenomena that have yielded promising results over the years. Of particular importance are GSA and PSO which are swarm-based meta-heuristics.

PSO was proposed by Ebelhart and Kennedy in 1995 (Eberhart and Kennedy, 1995). It simulates the social behavior of birds and fish. Its ability to efficiently solve numerous scientific and engineering optimization problems has given it increasing support and acceptance among researchers. Apart from optimization problems, PSO has been applied in feature selection (Xue et al., 2012; Chuang et al., 2008) as well as data clustering (Van Der Merwe and Engelbrecht, 2003). The algorithm has been used in the field of electromagnetics (Robinson and Rahmat-Samii, 2004) and image segmentation (Omran et al., 2006; Feng et al., 2005; Puranik et al., 2009). PSO has also been modified to improve its convergence capabilities to create a Quantum based PSO (Jeong et al., 2010) as well as an adaptive version described in Zhan et al. (2009). Oppositional learning has also been used to improve PSO's exploration capability by preventing the particles from getting trapped in a local minimum in OBPSO (Wang et al., 2007).

GSA was proposed by Rashedi and Saryazdi in 2008 (Rashedi et al., 2009) for solving single-objective optimization problems. It is based

* Corresponding author.
E-mail addresses: kardevroop@gmail.com (D. Kar), manosij1996@gmail.com (M. Ghosh), ritamguha16@gmail.com (R. Guha), raamsarkar@gmail.com (R. Sarkar), ir1gahel@uco.es (L. Garcia-Hernandez), ajith.abraham@ieee.org (A. Abraham).

https://doi.org/10.1016/j.engappai.2020.103847
Received 16 September 2019; Received in revised form 5 June 2020; Accepted 21 July 2020
Available online 4 August 2020

on Newtonian gravity stating "Every particle in the universe attracts every other particle with a force which is directly proportional to the product of their masses and inversely proportional to the square of the distance between the masses". Variations of GSA include QIGSA (Quantum Inspired GSA) (Soleimanpour-Moghadam et al., 2014) which uses quantum mechanics theories to prevent the premature convergence problem of GSA. CGSA (Li et al., 2012) combines chaos with GSA for selection of parameters using chaos theory. A binary version of the GSA has also been developed called BGSA (Rashedi et al., 2010) by the same authors, Rashedi and Saryazdi, that efficiently tackles feature selection and dimension reduction (Papa et al., 2011). This algorithm has also been combined with *Simulated Annealing* to form GABSA (Gravitation Algorithm Based Simulated Annealing) (Tong, 2014). GSA has been modified and used for various other real-world problems like data clustering (Yin et al., 2011). GSA has also been implemented in the optimization of power despatch in a grid (Soleimanpour-Moghadam et al., 2014; Duman et al., 2012), forecasting turbine heat rate (Zhang et al., 2013) and also in image segmentation (Sun and Zhang, 2013).

Hybrid algorithms combining PSO and GSA have also been proposed in the literature. These include PSOGSA (Mirjalili and Hashim, 2010) which integrates the ability of exploitation in PSO with the exploration ability in GSA to harness both algorithms' strengths. Comparison of the hybrid algorithms with both the standard PSO and GSA algorithms by testing against some benchmark functions shows that the hybrid algorithm has a better capability to escape from local optima with faster convergence rate than the standard PSO and GSA. Another such hybrid algorithm is the Gravitational Particle Swarm (GPS) (Tsai et al., 2013) in which a GPS agent has attributes of both GSA and PSO. GPS agents update their respective positions with PSO and GSA velocities. GPS agents, therefore, can exhibit both social and cognitive behavior, and motion of birds in flight as shown by the PSO algorithm (Eberhart and Kennedy, 1995) along with the law of gravity utilized in GSA (Rashedi et al., 2009). Results show that both GPS and PSOGSA outperform both PSO and GSA by a significant margin. Therefore, we choose these two hybrids of GSA and PSO as our base algorithm for further improvement.

But the main limitation of these hybrid algorithms is that they have poor local search capabilities especially in GPS which is pointed out in Angeline (1998). Due to their fast convergence rate, they suffer from premature convergence. So, the algorithm may, in most cases, converge to some local optima (sometimes very close to the global optima) and is stuck there which hinders the achievement of the best result, like in the case of their ancestors. To get rid of this problem, the concept of an exploratory operator called centroid based fuzzy mutation has been introduced in GPS and PSOGSA. This addition of mutation allows us to address the problem of premature convergence, which is the main contribution of this paper. We have tested the proposed algorithms on a set of benchmark functions and the results corroborate our assumption.

Apart from the algorithms mentioned previously, we have also compared our results to algorithms like Opposition-based PSO (OBPSO, 2007) (Wang et al., 2007), which was a step in the direction to address the tendency of PSO to get trapped in a local optima, as well as, Sine Cosine Algorithm (SCA, 2016) (Mirjalili, 2016), Opposition-based SCA (OBSCA, 2017) (Abd Elaziz et al., 2017), Social Spider Algorithm (SSO, 2013) (Cuevas et al., 2013), League Championship Algorithm (LCA, 2009) (Kashan, 2009), Soccer League Competition Algorithm (SLC, 2014) (Moosavian and Roodsari, 2014) and Volleyball Premier League Algorithm (VPL, 2018) (Moghdani and Salimifard, 2018). These algorithms are outperformed in almost 70% cases by the proposed algorithms.

Engineering design problems involve defining values of design parameters which gives the best output for a mechanical device, structure, or system. This process for determining the best values is called engineering optimization. Sometimes many variables need to be adjusted while satisfying several conflicting objectives and/or constraints. Therefore, implicitly determining values using intuition becomes very difficult. Optimization using evolutionary algorithms can play a huge role here. There are several works on the use of evolutionary algorithms in engineering design problems (Dasgupta and Michalewicz, 2013; Deb, 1999). To portray the usefulness of our algorithms, we have applied them to five benchmark engineering design problems (Kohli and Arora, 2017) — Tension/Compression Spring Design problem, Gear train design problem, Welded Beam Design problem, Pressure design vessel problem and Closed coil helical spring design problem.

The contributions of this manuscript are presented below:

i. Development of an effective fuzzy-based mutation for hybrids of PSO and GSA namely GPS and PSOGSA to solve the problem of premature convergence and improve their local searching capabilities.
ii. Evaluation of the proposed algorithms on several benchmark functions to prove the effectiveness and relative superiority of the same.
iii. Application of the algorithms on some classical engineering design problems to show their practical usage.

2. Methods and methodologies

The hybrid algorithms we consider in our work — PSOGSA and GPS are described briefly in Sections 2.1 and 2.2 respectively. It should be noted that the points in our search space are referred to as points, particles and agents inter-changeably and refer to the same.

2.1. Hybrid Particle Swarm and Gravitational Search Algorithm (PSOGSA)

PSOGSA (Mirjalili and Hashim, 2010) is a novel hybrid optimization algorithm, combining the strengths of both PSO and GSA. It has been shown through results that this algorithm outperforms both PSO and GSA in terms of exploration and exploitation. The original version of this algorithm is well suited for problems with continuous search space.

The basic idea of PSOGSA is to combine the ability of social thinking ($gbest$) in PSO with exploration capability of GSA. The PSOGSA algorithm was mathematically modeled as similar to PSO and GSA where every search agent has a position vector reflecting the current position in search spaces as follows:

$$X_i = \left(x_{i1}, \ldots, x_{id}, \ldots, x_{iD}\right), i = 1, 2, \ldots, N \tag{1}$$

N is the number of search agents, d is the index and D is the dimension of the problem, and x_{id} is the position of the ith agent in the dth dimension. Optimization process begins with filling out the position matrix with random values. During optimization, the gravitational force from agent j on agent i at a specific time t is defined as follows:

$$F_{ij}^{d} = G(t) * \left(\frac{M_{pj}(t) * M_{aj}(t)}{R_{ij}(t) + \varepsilon}\right) * \left(x_{jd}(t) - x_{id}(t)\right) \tag{2}$$

M_{aj} is the active gravitational mass related to agent j, M_{pi} is the passive gravitational mass related to agent i, $G(t)$ is a gravitational constant at time t, ε is a small constant, and $R_{ij}(t)$ is the Euclidian distance between two agents i and j at time t. We consider the values of the two gravitational masses to be the same ($M_{pj} = M_{aj}$).

Gravitational and inertial masses are simply calculated by the fitness evaluation. A heavier mass means a fitter agent (corresponds to a lower value). This means that the better agents have higher attraction and move more slowly. Assuming the equality of gravitational mass and inertial mass, values of masses are calculated using the value of fitness. The gravitational and inertial masses are updated by the following equations:

$$M_{ai} = M_{pi} = M_{ii} = M_i, i = 1, 2, \ldots, N \tag{3}$$

M_{ii} is the inertial mass of the ith agent and M_i is the overall mass of the ith agent.

$$m_i = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{4}$$

$$M_i = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{5}$$

$fit_i(t)$ represents the fitness value of agent i at time t, and $worst(t)$ and $best(t)$ are defined as follows (for a minimization problem):

$$best(t) = \min_{j\in\{1,\ldots,N\}} fit_j(t) \tag{6}$$

$$worst(t) = \max_{j\in\{1,\ldots,N\}} fit_j(t) \tag{7}$$

It is to be noted that for a maximization problem, *max* is used in place of *min* and vice versa in Eqs. (6) and (7) respectively.

G and R_{ij} between two agents i and j are calculated as follows:

$$G(t) = G_o * exp(-\alpha * iter/maxiter) \tag{8}$$

$$R_{ij} = \sqrt[2]{\sum_{k=0}^{D} \left(X_{ik}(t) - X_{jk}(t)\right)^2} \tag{9}$$

α is the descending coefficient, G_0 indicates the initial gravitational constant, *iter* is the current iteration, and *maxiter* is the maximum number of iterations. In a problem space for the dth dimension, the total force that acts on agent i is calculated by the following equation:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \tag{10}$$

$rand_j$ is a random number generated with uniform distribution in the interval [0, 1]. The law of motion has also been utilized in this algorithm which states that acceleration of a mass is proportional to the resultant force and inverse of its mass, so the acceleration of all agents is calculated as follows:

$$a_{id}(t) = \frac{F_{id}(t)}{M_{ii}(t)} \tag{11}$$

M_{ii} is the inertial mass of agent i. During optimization, the best-obtained solution so far is saved as *gbest* following the concept of PSO. Eq. (12) was proposed as follows for combining PSO and GSA:

$$V_i(t+1) = rand * V_i(t) + c_1 * a_i(t) + c_2 * \left(gbest - X_i(t)\right) \tag{12}$$

$V_i(t)$ is the velocity of agent i at time t, c_j is an accelerating factor, *rand* is a random number generated with a uniform distribution between 0 and 1, $a_i(t)$ is the acceleration of agent i at time t, and *gbest* is the best-obtained solution so far. In each iteration, the positions of agents are updated as follows:

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{13}$$

In PSOGSA, at first, all the agents are randomly initialized using uniform distribution. Each agent is considered as a candidate solution. After initialization, F_{ij}^d, $G(t)$, and $F_i^d(t)$ are calculated by Eq. (2), Eq. (8) and Eq. (10) respectively. Whereas, the accelerations of particles are defined by Eq. (11). In each iteration, the best-attained solution should be updated. After calculating the acceleration and updating the best solution, the velocity of each agent is calculated by Eq. (12). Finally, the positions of agents are updated by Eq. (13). The process of updating velocities and positions is stopped when an end criterion is met.

2.2. Gravitational Particle Swarm (GPS)

GPS (Tsai et al., 2013) is a swarm intelligence based hybrid algorithm which incorporates both the ideas of PSO and GSA. The particles under consideration in GPS update their respective positions based on both PSO and GSA velocities. Thus, GPS, as a whole, exploits both the social behavior of PSO as well as the population-based search pattern of GSA. Population and the respective particle positions and velocities are represented in the following manner.

$$X_i = \left(x_{i1}, \ldots, x_{id}, \ldots, x_{iD}\right), i = 1N \quad \text{where } D \text{ is the dimension} \tag{14}$$

$$v_i^d(t+1)_{PSO} = w(t)\, v_i^d(t) + c_1 r_{i1}\left(pbest_i^d - x_i^d(t)\right) + c_2 r_{i2}\left(gbest_i^d - x_i^d(t)\right) \tag{15}$$

$$v_i^d(t+1)_{GSA} = rand_i * v_i^d(t) + F_i^d(t)/M_i^d(t) \tag{16}$$

$$v_i^d(t+1)_{GPS} = c_3 r_{i3} * v_i^d(t+1)_{PSO} + c_4\left(1 - r_{i3}\right) * v_i^d(t+1)_{GSA} \tag{17}$$

$$v_i^d(t+1) = x_i^d(t) + v_i^d(t+1)_{GPS} \tag{18}$$

Eq. (15) and Eq. (16) have been taken from PSO formulation (Rashedi et al., 2009) and GSA formulation (Eberhart and Kennedy, 1995) respectively, while Eq. (17) is the GPS velocity update based on Eq. (15) and Eq. (16). Of which, r_{i3} is a random variable lying uniformly within [0, 1] to create stochastic impacts of PSO velocity and GSA velocity on GPS agent positions. c_3 and c_4 are two constants to determine the degree to which PSO and GSA velocities influence GPS. GPS is defined as GPS (N, c_3, c_4). When both of c_3 and c_4 are valued at 1, GPS agents are stochastically impacted by equal influences of PSO and GSA.

2.3. Fuzzy logic

Since inception in 1965 through the fuzzy set theory concept by Zadeh (1996), it has been widely used and applied to a variety of fields like cancer classification (Schaefer et al., 2009), image segmentation (Lim and Lee, 1990), optimization (Liu and Lampinen, 2005) and so on. Most natural things cannot be defined by simple or convenient shapes or distributions. Fuzzy logic is the characterization of the truth value of a variable as a real number between $[0, 1]$. Membership Functions (MFs) are used to define the fuzziness in a graphical form for eventual use in fuzzy set theory.

3. Proposed model

The hybrid versions of PSO and GSA — PSOGSA and GPS suffer from premature convergence. This causes the algorithms to get trapped in a local optimum and it deters us to get the optimal solution. Hence, mutation has been applied to these hybrid versions of PSO and GSA, i.e. PSOGSA and GPS, to get rid of the problem of premature convergence. Due to the fast convergence nature of these hybrids, all the points tend to move fast towards the current best solution in each iteration. However, the point it converges to may not be the most optimal and to ensure that this does not occur, we propose a novel mutation approach called *centroid based fuzzy mutation*.

3.1. Fuzzy logic-based mutation

The concept of Fuzzy logic has widely been used for solving different research problems and it has applications from industry to academia. Following the concept of fuzzy logic, briefed in Section 2.3, an MF gives a corresponding membership value of an operation. Any fuzzy set F in the universal domain U can be defined as a collection of ordered pairs. The mathematical representation of such a set is provided below:

$$F = \{(x, \mu_F(x)) | x \epsilon U\}$$

Where μ_F is the MF of F with values in the range $[0, 1]$ and x is an element of information in universal set U. So, depending on the nature of the MF, an element of information can have different degrees of membership in the present domain. The elements having full membership form the core of the fuzzy set, the ones having non-zero membership are called support and the ones having non-zero but incomplete membership (< 1) are said to be the boundary of the fuzzy set.

Formation of an MF

The MFs in Fuzzy logic has a crucial role in the overall performance of the fuzzy representation of the underlying problem (Sadollah, 2018). To be specific, the shape of a MF is important for a particular problem as it takes the decisive role of the fuzzy inference system. MFs can be of different shapes — Gaussian, triangular, trapezoidal etc. with the condition that the values of a MF vary from 0 to 1. An MF basically maps the given data with required degree of memberships. Deep understanding about the underlying problem can give us notion to know which shape of MF would fit the application under consideration.

There may be infinite number of ways to characterize fuzziness. The choice of which depends on the problem type. Therefore, apart from shape of a MF, deciding the interval as well as number of MFs is very important. For example, to model a control system in terms of temperature by fuzzy logic, it is vital to know how many MFs are required (e.g., high, medium, and low) along with the interval of membership values. These two parameters have a significant impact on the inference of a fuzzy logic-based system. Besides, observing the data distribution is another important factor. Many times, trial and error methods are applied for selecting the shape of MF shape as there is no exact method for selecting the MFs. The function may have an arbitrary curve, and it suits us in terms of efficiency, simplicity, and speed.

However, the number of MF has greater influence as it determines the computational time. Hence, the optimum model can be determined by varying the number/type of MFs for achieving best system performance. The work reported in Mitaim and Kosko (1996) gives some idea about the shape which would be best if someone applies fuzzy logic as a universal approximator. In another work (Chen and Otto, 1995), a constrained interpolations concept was designed to fit a MF to a finite number of membership values. Some other works are found in the literature giving some directions of choosing MF (Wu, 2012; Ross, 2005; Rutkowska, 2016; Czekalski, 2006). The main concern is to break the 0–1 modeling, and it can be done by applying a triangular MF. Nevertheless, if the situation is more complex, we may require special type of MF. To make the best choice, a high-fidelity intuition based on adequate experience can give a satisfactory answer.

Using metaheuristic optimization methods and evolutionary optimization algorithms, fuzzy logic possesses the great flexibility toward its initial parameters regarding MFs (El-Zonkoly et al., 2009). Interested reader can find some useful information about MFs and some procedures (e.g., GA and neural network) to assign memberships to fuzzy variables (Ross, 2005).

This concept of fuzzy logic has been used for finding the probability of mutation being performed for a particle. At any moment, mutation is not completely certain or uncertain for a particle, instead, the membership value provides the probability of mutation. The proposed mutation-based model has been detailed in Section 3.2. Thus, incorporation of fuzziness allows us to perform mutation in PSOGSA and GPS in a probabilistic manner.

3.2. Centroid based fuzzy mutation

This newly developed mutation helps the particles in the population to drift when there is the chance to pre-mature convergence. We have the following two important metrics when we consider a particle for mutation.

- The overall distance of the particles from the other particles
- History of the particles i.e. the change in the accuracies of the particles as a whole

First consideration checks the distance of a particle from the centroid of population. If the particles come closer to each other (overall distance among them decreases), then there is a possibility of premature convergence. To avoid this, we apply mutation to place the particles away from each other. This helps the particles to circumvent the local optimum and look for some other region in the search space. Instead of calculating the distance of a particle from every other particle, it is convenient to measure the distance of the particle from the centroid of all the particles. If this distance is less, it implies that the particle is close to the other particles and hence should be considered for mutation. So, we can see that the probability of mutation is inversely proportional to the distance of the particle from the centroid. In certain situations, it may so happen that a particular particle is residing at the centroid. This will make the distance to be 0 leading to infinite chances of mutation which is unacceptable. That is why we need to add 1 to the distance to ensure that this scenario never occurs. So, the contribution of distance to the probability of mutation is presented in Eq. (19) where $dist$ is the distance of the particle from the centroid and P_d is the estimated contribution.

$$P_d = \frac{1}{1 + dist} \tag{19}$$

Similar to distance, history of the particles may provide some important insight into the mutation probability. When the global best particle gets changed frequently over the iterations, it indicates that the particles are still exploring and trying to reach better solutions in the search space. But, on the other hand, if the global best particle is static over a significant number of iterations, it gives an indication that the particle might have gotten stuck in some local optima and is unable to explore different parts of the search space. In these scenarios, it becomes important to perform mutation on the particles to provide some perturbation to them, and thereby helping the particles to overcome the local optima and to reach the global optima. Thus, the probability of mutation should increase when the time for which the global best particles remains constant increases. We estimate the contribution of this historical information (P_c) to the probability of mutation using Eq. (20) where $unchanged$ is the number of iterations for which the global best particle has remained unaffected. We take $\alpha = 4$ and $\beta = 5$. Following these values, the probability of mutation increases as the value of $unchanged$ increases.

$$P_c = a + b * tanh\left(\left(\frac{unchanged}{\alpha}\right) - \beta\right) \tag{20}$$

where $a = 0.5$ and $b = 0.5$. Depending on the value of the $unchanged$, the hyperbolic tan function will return a value in $[-1, 1]$ which when multiplied by b will be restricted in the range of $[-0.5, 0.5]$. So, ultimately the value of P_c is in $[0, 1]$. We combine these two contributions using Eq. (21). Thus, if the $gbest$ is moving towards the optimal solution and the distance becomes large from the centroid then the point may explore an uncharted portion of the search space and hence the motion of the point is not disturbed. On the other hand, if the point goes closer to the centroid and $gbest$ remains unchanged for long duration then the point approaches to a well-explored portion of the search space, and therefore mutation is applied to disrupt its movement in order to explore a different region of the search space. The parameters ρ and φ assign weight to the contributions of distance and history in the probability equation. We have used ρ as 0.6 and φ as 0.4. Distance has been given more importance over history as there may be certain cases where although $gbest$ does not change, other particles change their positions. In this scenario, the particles do not get stuck but the approach will consider a probable convergence. So, to avoid that, we have assigned a lesser weight to history.

$$P_i = \rho * P_d + \varphi * P_c \tag{21}$$

P_i denotes the probability of mutation for the ith particle of population. If P_i is greater than a generated random value, the particle gets mutated, else no mutation takes place. In the proposed model, the term P_i acts as membership value for fuzzy mutation. Mutation is not applicable for every particle. Instead P_i helps us to find a set of fuzzy elements similar to the fuzzy set denoted as F in Section 2.3. After obtaining this set of fuzzy particles, if for particle i, P_i is greater than

a generated random value, the particle gets mutated, else no mutation takes place. We perform mutation using the following functions:

$$\Delta q = 0.5 * range * \left(\left(1 - \frac{count}{iter}\right)^2\right) \tag{22}$$

$$\Delta p = \min(\Delta q, P_{ij}) \tag{23}$$

where Δp is the change in jth dimension's value of the particle, *range* is the difference between the upper and lower limits of the domain of the benchmark function under consideration, *count* denotes the current iteration number and *iter* is the total number of iterations to be performed and P_{ij} is the value of the ith particle in the jth dimension of the entire population. The value of Δq gradually decreases over time to allow less disruption as the points converge. Δp is restricted (in Eq. (23)) to ensure the disruption in the motion of an agent is limited.

Mutation occurs alternatively subtracting or adding Δp to X_{ij} with the probability of addition and subtraction being half. This allows the point to move by a value of Δp in any direction in the D dimensional space. Δp is evaluated D times as well as performed subtraction or addition of the value Δp to each of the X_{ij} for all $j \in 1D$.

Trivially the importance of the fuzzy mutation is described in Fig. 1(A–D). Consider the scenario when there are two local minima — one having lesser value (desirable) than the other for a minimization problem. The particles are moving according to the motion defined by PSOGSA or GPS. There is a high chance that the particles will converge to a local minimum without even considering the other one. Fig. 1A represents the force diagram of three particles M1, M2 and M3 when they are closer to local minimum 1. The progression of the particles is shown in Fig. 1B where they are almost converged to local minimum 1. To avoid further convergence, fuzzy mutation is used. Say only M1 and M2 pass the mutation criteria. The mutation direction of both particles are shown in Fig. 1C. Depending on the extent of mutation, the particles may land into the final state described in Fig. 1D where M1 and M2 have successfully avoided local minimum 1 and moved towards local minimum 2. We know that there are a lot of assumptions regarding this scenario but if we compare the GPS or PSOGSA with their mutated versions, the latter ones will always have better chances of avoiding such convergence problems.

4. Experimental results

The experiments in this work were performed on MATLAB in a PC having 4GB RAM. The proposed fuzzy mutation-based hybrid versions of PSOGSA and GPS, MPSOGSA and MGPS respectively, have been tested on several benchmark functions as given in Tables 1–3 in the following section. The results obtained for these benchmark functions are provided in Section 4.1 while graphical depictions of convergence of the points are shown in Section 4.2. The computation complexity of the fuzzy mutation algorithm is low and so our proposed algorithm has the same complexity as GPS and PSOGSA.

Three categories of functions used in Rashedi et al. (2009) are adopted to test MGPS and MPSOGSA. These categories include seven unimodal high-dimensional functions (F_1–F_7 in Table 1); six multimodal high-dimensional functions (F_8–F_{13} in Table 2), and ten multimodal low-dimensional functions (F_{13}–F_{23} in Table 3). Usually, the optimization of unimodal functions focus on the algorithm's rate of convergence to a global optimum. On the other hand, due to the presence of multiple functional valleys, multimodal functions test the ability of the algorithm to cricumvent local optima to reach the global optimum. So, we can say, in a way, unimodal functions help to evaluate exploitation and multimodal functions investigate exploration abilities of the algorithm under consideration.

4.1. Results on benchmark functions

This section illustrates the results obtained by the proposed algorithms over 23 benchmark functions from unimodal, multimodal and multimodal with fixed dimension categories. The final results are also compared with some state-of-the-art algorithms to justify the applicability of the proposed mutation models.

Table 1
Description of unimodal functions used in present work.

Function	Domain	Optimum	Position
$F_1 = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^{30}$	0	$(0)^{30}$
$F_2 = \sum_{i=1}^{n} x_i \vee + \prod_{i=1}^{n} x_i \vee$	$[-10, 10]^{30}$	0	$(0)^{30}$
$F_3 = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]^{30}$	0	$(0)^{30}$
$F_4 = max\left\{\lvert x_i \rvert, 1 \le i \le n\right\}$	$[-100, 100]^{30}$	0	$(0)^{30}$
$F_5 = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right]$	$[-30, 30]^{30}$	0	$(1)^{30}$
$F_6 = \sum_{i=1}^{n} \left(x_i + 0.5\right)^2$	$[-100, 100]^{30}$	0	$(0)^{30}$
$F_7 = \sum_{i=1}^{n} i x_i^4 + random[0, 1]$	$[-1.28, 1.28]^{30}$	0	$(0)^{30}$

4.1.1. Parameter tuning

In order to obtain proper results of the proposed models, some experimentations have been performed to fine-tune the parameters present in the algorithms. There are mainly four parameters — α, β as given in Eq. (20) and ρ, φ as given in Eq. (21). Apart from these four parameters, there are number of iterations and population size which should have a fixed value for a uniform testing environment. For all the experimentations and comparisons, we have fixed the population size to be 50 and used 500 iterations for F_1–F_7, 1000 iterations for F_8–F_{23}. For the previously mentioned four parameters, the qualities of the solutions are checked by varying their values and finally the most optimal combination out of them is selected.

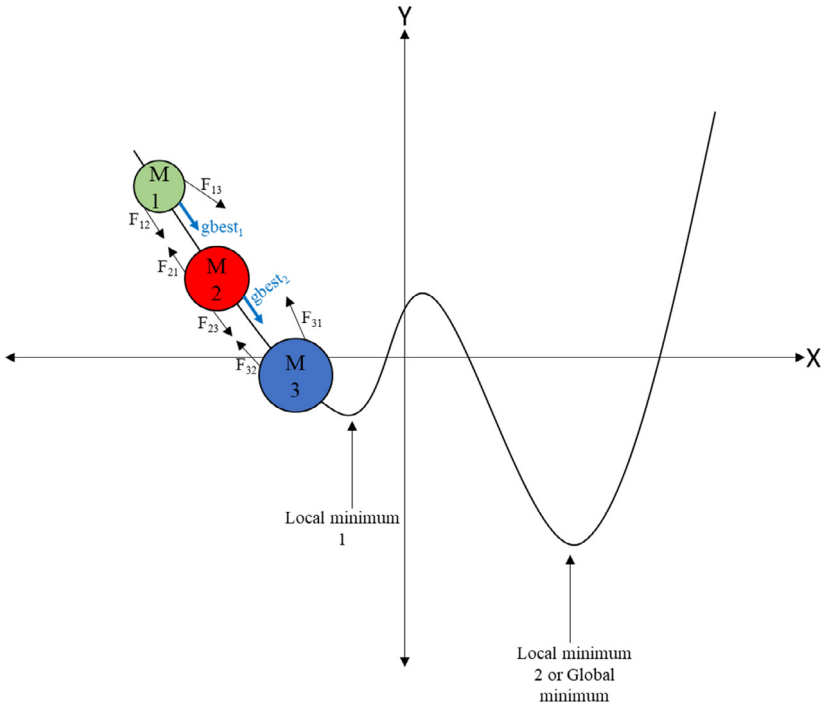
In order to select the optimal values of the parameters, one function from each category of the benchmark functions has been selected — F_1 from the set of unimodal functions, F_{10} from the set of multimodal functions, and F_{15} from the set of multimodal functions with fixed dimension category. The testing is done for MGPS algorithm. At first, α, β values of the model have been varied and tested on these three functions followed by the testing of ρ, φ values over the same three functions. Graphical representations of the results obtained through the testing are provided in Fig. 2.

After testing, the final values for the parameters are selected as mentioned in Table 4. For rest of the experimentations, these values have been used.

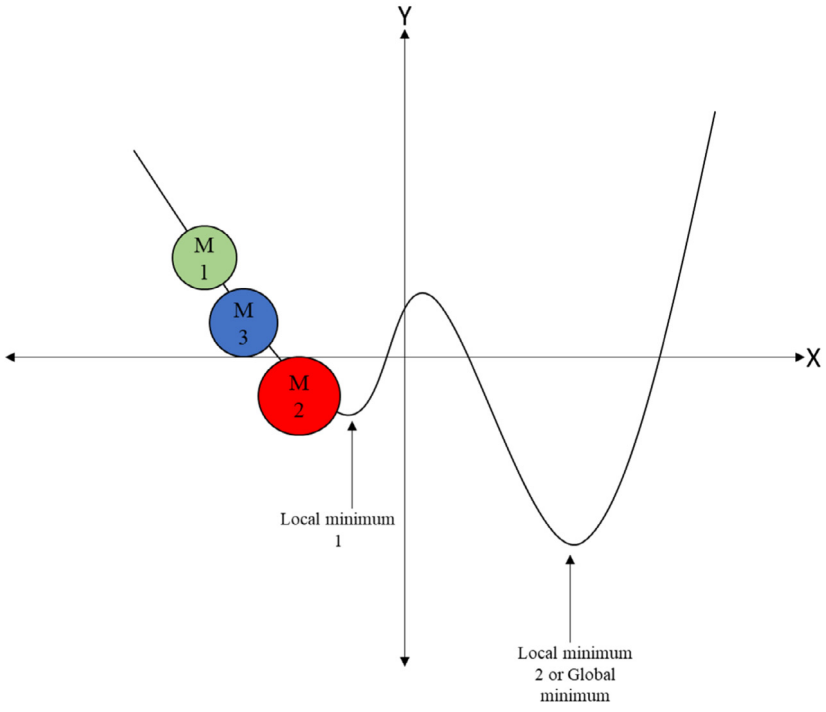
4.1.2. Comparison with state-of-the-art

The results of our two optimization models have been tabulated in Tables 5–7 against four other optimization approaches namely, PSO, GSA, PSOGSA and GPS. PSO is used to represent PSO (50); the value 50 signifies the population size, i.e., the number of population points. GSA represents GSA (50); GPS represents GPS (50, 1.0, 1.0) with equal influences from PSO and GSA. PSOGSA as well as our proposed models, MGPS and MPSOGSA use the same parameter values. All the used algorithms have been run independently for 25 times out of which only the best 20 results have been used to measure the average, best, worst and standard deviation (Sd.) values. The least value obtained for the corresponding fitness function is taken to be the better result here. The performance of each algorithm has been determined by comparing the average fitness value given by the corresponding algorithm for each function and then, if need be, the best (minimum) value returned by an algorithm for that function and then the standard deviation (Sd.) value (if the average and best values are same). The most optimum value has been bolded and underlined. The second-best value has been only bolded (not underlined). MGPS outperforms GPS 13 out of 23 times (56.52%) and MPSOGSA outperforms PSOGSA 17 times out of 23 (73.91%).

We have further used the results of contemporary optimization algorithms including SSO (Social Spider Optimization, 2013) (Cuevas et al., 2013), SLC (Soccer League Championship Algorithm, 2014) (Moosavian and Roodsari, 2014), SCA (Sine Cosine Algorithm, 2016) (Mirjalili, 2016), OBSCA (Opposition-based SCA, 2017) (Abd Elaziz et al., 2017) and VPL (Volleyball Premier League Algorithm, 2018) (Moghdani and Salimifard, 2018) for comparison. The fitness values for the algorithms



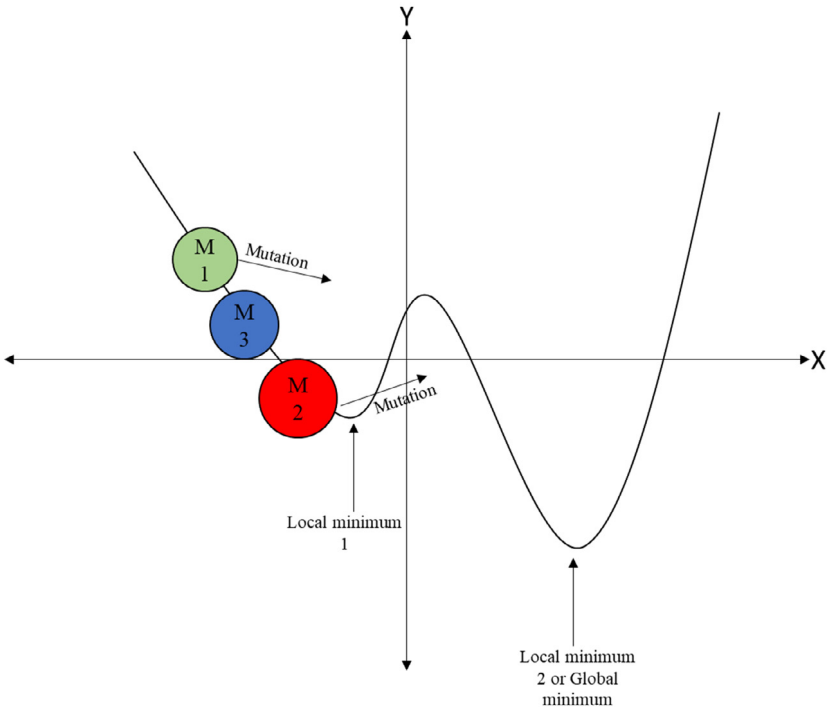
There are two minima in the graph - local minimum 2 (may be global minimum) is having lesser value than local minimum 1. Force diagram of 3 particles namely M1, M2 and M3 are shown when they are near local minimum 1.



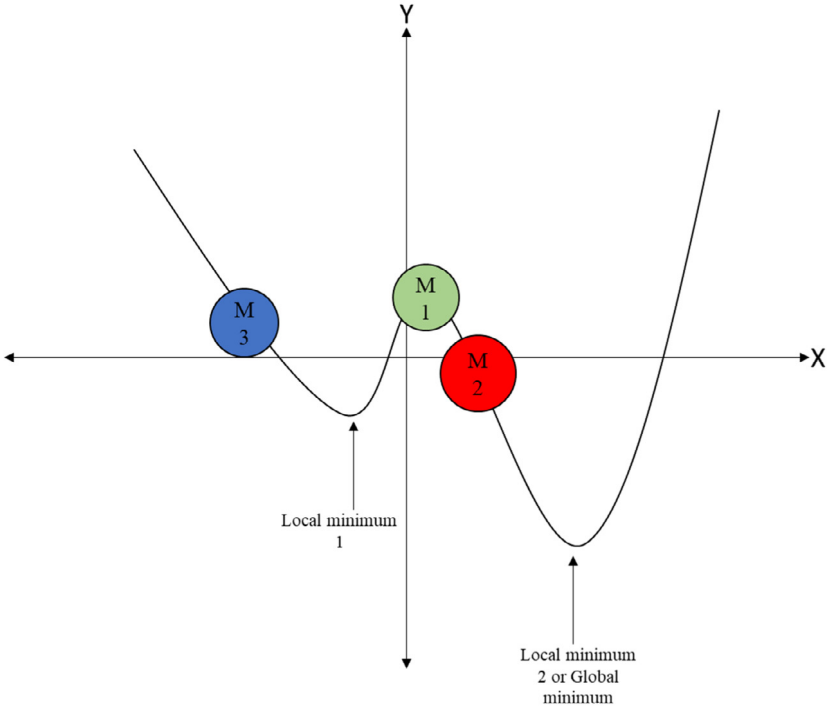
The particles are getting converged to local minimum 1 following the movement defined in PSOGSA (using Eqn. 1-13) or GPS (using Eqn. 14-18).

Fig. 1. (A–D) Example illustrating the utility of mutation in PSOGSA and GPS to avoid convergence to local minima.

SCA (Sine Cosine Algorithm), OBSCA (Opposition-based SCA), OBPSO (Opposition-based PSO) and SSO (Social Spider Optimization) have been taken from the paper on OBSCA (2017) (Abd Elaziz et al., 2017) to construct Tables 8–10 and those of Tables 5–7 for LCA (League Championship Algorithm) and SLC (Soccer League Competition Algorithm) have been taken from the paper on VPL(2018) (Moghdani and Salimifard, 2018). Our algorithms collectively outperform OBPSO in over 85% cases, SCA in over 85% cases, SSO in nearly 70% cases, OBSCA in 70% cases. The values for the algorithms VPL, LCA, SLC have been referred from the paper on VPL (2018) (Moghdani and Salimifard, 2018). It can be seen that our algorithms collectively outperform SLC in over 75% cases, LCA in nearly 60% cases and VPL in 56.52% cases.



Depending on the probability value presented in Eqn. 21, say M1 and M3 are getting mutated while M2 is not. The extent of the mutation is caluclated using Eqn. 23.



After mutation of particles M1 and M2, they have succesfully circumvented local minimum 1 and started moving towards local minimum 2

Fig. 1. (*continued*).

The results in Tables 6–7 show that the proposed algorithms perform well in the category of multimodal functions. Out of 16 functions, the proposed algorithms outperform the others in 8 functions, which show that the use of mutation has allowed the algorithms to avoid being stuck in local optima. It should be noted that in case of unimodal functions as well, the proposed algorithms perform quite well in comparison to their parents GPS and PSOGSA. In total, the proposed algorithms are the best in 9 cases and the second best in 4 cases. In comparison, VPL is best in 6 cases and second best in 4 cases. LCA on the other hand is best in 5 cases and second best in 2 cases. This shows that, rank-wise, the proposed algorithms perform quite well.

Table 2
Description of multimodal functions used in present work.

Function	Domain	Optimum	Position
$F_8 = \sum_{i=1}^{n} -x_i sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500, 500]^{30}$	−12569.5	$(420.96)^{30}$
$F_9 = \sum_{i=1}^{n} \left[x_i^2 - 10cos\left(2\pi x_i\right) + 10\right]$	$[-5.12, 5.12]^{30}$	0	$(0)^{30}$
$F_{10} = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos\left(2\pi x_i\right)\right) + 20 + e$	$[-32, 32]^{30}$	0	$(0)^{30}$
$F_{11} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^{30}$	0	$(0)^{30}$
$F_{12} = \frac{\pi}{n}\left\{10sin^2\left(\pi y_i\right) + \sum_{i=1}^{n-1}\left(y_i - 1\right)^2\left[1 + 10sin^2\left(\pi y_{i+1}\right)\right] + \left(y_n - 1\right)^2\right\} + \sum_{i=1}^{n} u\left(x_i, 10, 100, 4\right)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m, x_i > a \\ 0, -a \le x \le a \\ k\left(-x_i - a\right)^m, x_i \leftarrow a \end{cases}$	$[-50, 50]^{30}$	0	$(1)^{30}$
$F_{13} = 0.1\left\{sin^2\left(3\pi x_1\right) + \sum_{i=1}^{n}\left(x_i - 1\right)^2\left[1 + sin^2\left(3\pi x_i\right)\right] + \left(x_n - 1\right)^2\left[1 + sin^2\left(2\pi x_n\right)\right]\right\} + \sum_{i=1}^{n} u\left(x_i, 5, 100, 4\right)$	$[-50, 50]^{30}$	0	$(1)^{30}$

Note: e is Euler's constant.

Table 3
Multimodal functions with fixed dimension used in present work.

Function	Domain	Optimum	Position
$F_{14} = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}\left(x_i - a_{ij}\right)^6}\right)^{-1}$	$[-65.53, 65.53]^2$	1	(−32,32)
$F_{15} = \sum_{i=1}^{11}\left[a_i - \frac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	$[-5, 5]^4$	0.00030	(0.1928,0.1908,0.1231,0.1358)
$F_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$	−1.0316	(0.089,0.712),(−0.089,0.712),
$F_{17} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	$[-5, 10] \times [0, 15]$	0.398	(−3.14,12.27), (3.14,12.275), (9.42,2.42)
$F_{18} = \left[1 + \left(x_1 + x_2 + 1\right)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2\right)\right] \times \left[30 + \left(2x_1 - 3x_2\right)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2\right)\right]$	$[-5, 5]^2$	3	[0, −1]
$F_{19} = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	$[0, 1]^3$	−3.86	(0.114, 0.556, 0.852)
$F_{20} = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	$[0, 1]^6$	−3.32	(0.201, 0.15, 0.477, 0.275, 0.311, 0.657)
$F_{21} = -\sum_{i=1}^{5}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	$[0, 10]^4$	−10.1532	$5a_{ij}$
$F_{22} = -\sum_{i=1}^{7}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	$[0, 10]^4$	−10.4028	$7a_{ij}$
$F_{23} = -\sum_{i=1}^{10}\left[\left(X - a_i\right)\left(X - a_i\right)^T + c_i\right]^{-1}$	$[0, 10]^4$	−10.5363	$10a_{ij}$

Note: For detailed description of the functions of Table 3 refer to Appendix A of Rashedi et al. (2009).

Table 4
Values of different parameters used in the proposed models.

Parameter	Value
Population size	50
No. of iterations	500 iterations for F_1–F_7, 1000 iterations for F_8–F_{23}
α	4
β	5
ρ	0.6
φ	0.4

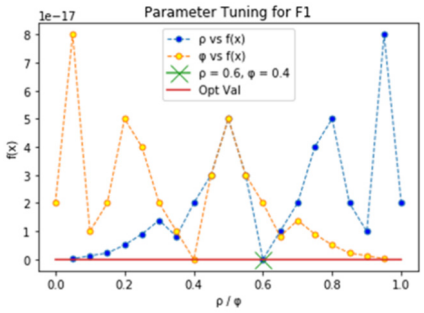
In comparison with OBSCA, SCS, OBPSO and SSO in terms of only best, average and standard deviation, it can be seen that proposed algorithms have performed quite well as well. The proposed algorithms are best in 10 cases and second best in 12 cases. So, in all, except 2 functions the proposed algorithms have rank of 1 or 2.

The proposed algorithms are very capable of avoiding local minima and thus perform quite well for the category of multimodal functions. The mutation, though in some cases, causes fluctuation as seen in Section 4.2 because it causes the particles to move in all directions which results into both increases and decreases in fitness value. The main applicability of the solution is for problems whose fitness function corresponds to multimodal functions of fixed dimension. The No Free lunch theorem (Wolpert and Macready, 1997) points out the fact that no one algorithm can outperform all others in all cases and this is what keeps research alive in this field. However, the better performances in comparison to other algorithms in most cases shows the applicability and effectiveness of the proposed algorithms.

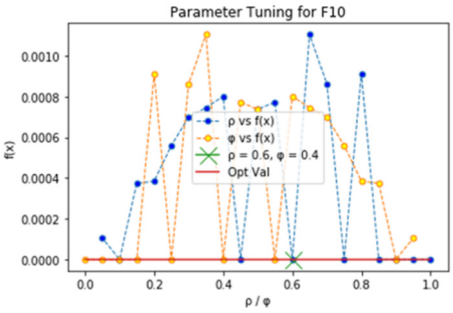
4.2. Graphical depiction of the results

We use graphs to show the convergence of our algorithms for some of the benchmark functions considered here. These graphs are plotted using the values for the best-obtained fitness value in the population for each iteration of the algorithm. For all the plots, *gbest* refers to the global best fitness value for all members in the population. For F_{15}, 0.00030 is added to the fitness result before plotting and for F_{19}, 3.8774 is added. We have shown one graph from each of the three categories of functions.

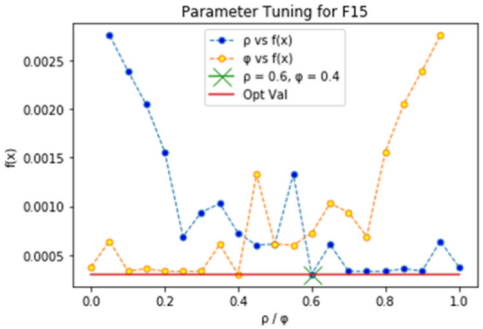
Fig. 3(A, C and E; B, D and F) shows the variation of the *gbest* value with the iterations for MGPS and MPSOGSA respectively. *gbest* denotes the fitness of the best particle of the population. The graphs show the convergence of the population. The spikes are due to mutation which causes variations in the global fitness to prevent the method from getting stuck in local optima. Some functions have wide fluctuation in performance like F_9 and F_{19}. It should be noted that in Tables 8–10 the performance of the proposed algorithm for F_{19} is the best and for



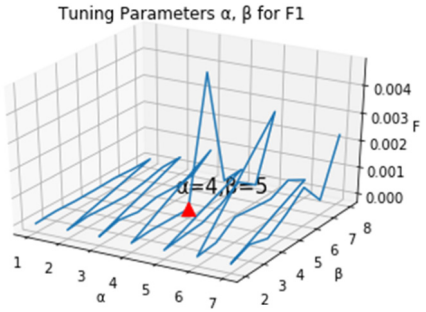
A: Graph showing the function value plotted against the parameter ρ/ϕ for F_1



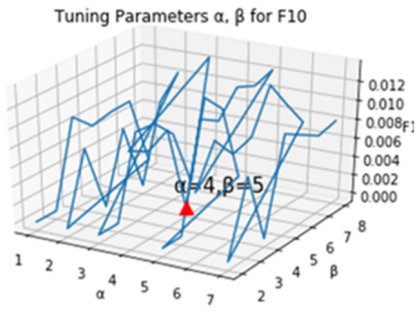
B: Graph showing the function value plotted against the parameter ρ/ϕ for F_{10}



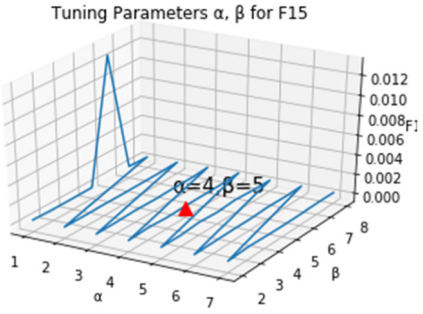
C: Graph showing the function value plotted against the parameter ρ/ϕ for F_{15}



D: Graph showing the function value plotted against the parameters α, β for F_1



E: Graph showing the function value plotted against the parameters α, β for F_{10}



F: Graph showing the function value plotted against the parameters α, β for F_{15}

Fig. 2. (A–C) Graphical representations of the results obtained via varying ρ/ϕ for F1, F_9, F_{15}. (D–F) Graphical representations of the results obtained via varying α, β for F1, F_9, F_{15}.

F_9 is second best. Similarly, for Tables 5–7 the performance for F_{19} is second best for the proposed algorithms. So, it can be concluded that the fluctuation does not hamper performance but is rather helpful in avoidance of local minima and premature convergence.

4.3. Time requirement analysis

From the results and corresponding discussion provided in Sections 4.1 and 4.2, it is clear that the addition of mutation does help PSOGSA and GPS avoid premature convergence, thereby helping them

Table 5
Comparison of results of optimization algorithms on unimodal functions. The best, average and standard deviation of different algorithms are provided.

Function	Value heads	GSA	PSO	GPS	PSOGSA	LCA	VPL	SLC	**Proposed algorithms**	
									MGPS	MPSOGSA
F_1	Best	1.1E−17	1.1E−15	6.6E−19	3.29E−19	1.41E−48	**0.00E+00**	**1.90E−166**	4.12E−21	3.10E−13
	Avg.	2.0E−17	1.3E−11	1.2E−18	4.74E−19	3.25E−46	**7.81E−132**	**4.40E−160**	9.38E−19	1.92E−09
	Sd.	5.5E−18	8.8E−11	3.0E−19	8.04E−19	1.79E−46	**4.20E−131**	**3.91E−80**	1.57E−18	3.67E−09
F_2	Best	1.4E−08	4.4E−09	3.3E−09	2.47E−09	**4.58E−25**	**1.12E−102**	1.12E−125	2.38E−22	2.11E−14
	Avg.	2.4E−08	2.9E−06	5.2E−09	2.93E−09	**9.79E−25**	**1.13E−90**	8.85E−06	1.40E−20	1.09E−10
	Sd.	4.4E−09	1.3E−05	9.0E−10	2.64E−10	**1.49E−24**	**5.13E−90**	9.84E+03	2.21E−20	4.48E−10
F_3	Best	7.5E+01	1.9E+01	3.1E+00	2.92E+02	3.26E+03	**1.93E−33**	2.58E−25	**1.17E−06**	2.5E−02
	Avg.	2.3E+02	1.2E+02	9.7E+01	1.82E+03	1.12E+03	**8.16E−04**	2.11E−02	**5.57E−07**	9.98E−01
	Sd.	1.0E+02	7.5E+01	1.1E+02	4.82E+02	6.06E+03	**2.85E−03**	1.33E+01	**5.52E−06**	7.1E+01
F_4	Best	2.1E−09	1.4E−01	8.2E−10	1.3E+01	1.49E+00	**0.00E+00**	8.96E−30	**1.17E−06**	8.30E−05
	Avg.	6.4E−02	4.2E−01	1.3E+00	2.2E+01	5.09E−01	**1.54E−29**	3.07E−04	**5.52E−06**	3.19E−04
	Sd.	2.5E−01	1.9E−01	9.8E−01	3.89E+00	2.63E+00	**3.96E−29**	1.03E+00	**1.88E−06**	2.74E−04
F_5	Best	2.6E+01	2.5E+01	2.3E+01	1.6E+01	**9.86E−03**	2.58E+01	3.96E+01	**4.19E+00**	2.24E+01
	Avg.	2.8E+01	2.7E+01	2.6E+01	2.6E+01	**8.42E−01**	2.62E+01	3.00E+01	**2.45E+01**	2.51E+01
	Sd.	1.0E+01	8.4E+00	8.8E+00	2.5E+00	**7.15E−01**	2.76E−01	3.32E−01	**9.47E+00**	6.65E−01
F_6	Best	7.4E−18	8.3E−16	6.0E−19	**3.30E−19**	**0.00E+00**	1.82E−05	1.02E+01	4.79E−06	1.75E−05
	Avg.	1.9E−17	1.3E−12	1.2E−18	**5.05E−19**	**0.00E+00**	4.09E−04	1.05E+01	2.21E−02	1.26E−02
	Sd.	6.4E−18	7.1E−12	3.3E−19	**9.40E−20**	**0.00E+00**	5.33E−04	1.12E−01	2.39E−02	5.59E−02
F_7	Best	8.4E−03	1.7E−03	1.1E−03	1.5E−02	1.56E−02	**4.67E−05**	**3.43E−01**	8.51E−04	1.13E−04
	Avg.	2.8E−02	7.0E−03	3.1E−03	3.3E−02	9.48E−03	**1.93E−03**	**3.76E−06**	1.73E−02	1.6E−02
	Sd.	1.7E−02	2.5E−03	1.2E−03	9.3E−03	3.44E−03	**1.36E−03**	**1.60E+01**	1.01E−02	1.0E−02

Table 6
Comparison of results of optimization algorithms on multimodal functions. The best, average and standard deviation of different algorithms are provided.

Function	Value heads	GSA	PSO	GPS	PSOGSA	LCA	VPL	SLC	**Proposed algorithms**	
									MGPS	MPSOGSA
F_8	Best	−4.2E+03	**−1.0E+04**	−8.9E+03	**−8.85E+03**	−3.72E+03	−1.19E+112	−1.33E−25	−9.11E+03	−7.86E+03
	Avg.	−2.7E+03	**−9.0E+03**	−7.5E+03	**−8.09E+03**	−7.53E+02	−4.68E+90	−7.07E+03	−7.46E+03	−7.31E+03
	Sd.	4.7E+02	**5.2E+02**	7.7E+02	**4.71E+02**	2.21E+03	2.15E+111	1.24E+02	6.21E+02	4.76E+02
F_9	Best	9.0E+00	1.8E+01	9.0E+00	4.48E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.14E−13	1.29E−11
	Avg.	1.7E+01	4.1E+01	2.1E+01	7.42E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.47E+00	2.89E−07
	Sd.	4.3E+00	1.5E+01	6.1E+00	1.1E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.84E+00	1.06E−06
F_{10}	Best	2.2E−09	4.6E−09	5.4E−10	4.32E−10	**2.22E−14**	**8.88E−16**	7.08E−16	2.86E−11	5.62E−07
	Avg.	3.4E−09	9.1E−08	8.8E−10	5.07E−10	**4.93E−15**	**8.88E−16**	7.05E−14	1.42E−08	5.83E−05
	Sd.	4.1E−10	2.0E−07	1.3E−10	4.70E−11	**3.74E−14**	**9.86E−32**	1.86E−29	3.59E−08	5.28E−05
F_{11}	Best	2.0E+00	5.1E−15	0.0E+00	2.86E−06	1.28E−13	**0.00E+00**	**0.00E+00**	0.0E+00	6.79E−14
	Avg.	4.3E+00	1.2E−02	2.3E−02	2.33E−01	5.84E−03	**0.00E+00**	**0.00E+00**	8.49E−05	1.09E−11
	Sd.	1.6E+00	1.2E−02	3.0E−02	3.5E−01	2.65E−03	**0.00E+00**	**0.00E+00**	2.96E−04	1.75E−11
F_{12}	Best	6.2E−20	1.6E−18	4.7E−21	1.01E+00	**1.57E−32**	**1.11E−06**	0.00E+00	1.00E−03	2.66E−08
	Avg.	2.5E−02	1.5E−02	5.0E−02	4.46E+00	**1.09E−47**	**2.58E−05**	1.04E+01	1.9E−03	1.2E−02
	Sd.	6.1E−02	3.6E−02	1.3E−01	1.80E+00	**1.57E−32**	**1.74E−05**	8.46E−82	5.32E−04	2.4E−03
F_{13}	Best	1.22E−18	**9.9E−131**	3.75E−08	9.29E−20	**1.35E−32**	2.63E−05	7.61E−01	2.56E−02	5.93E−01
	Avg.	2.1E−18	**2.0E−31**	8.48E−02	2.2E−03	**5.47E−48**	4.18E−04	1.00E+00	4.33E−01	9.73E−01
	Sd.	5.0E−19	**4.3E−31**	8.0E−02	4.5E−03	**1.35E−32**	4.84E−04	3.09E−01	3.1E−01	1.88E−01

to search for a global optimum solution. But, how computationally expensive mutation operation is? To find that out, we have provided the time requirements of both the algorithms before and after adding mutation operation over the 23 benchmark functions. The graphical representations of the time requirements are shown in Fig. 4(A, B).

From Fig. 4, it is clear that the mutation operation requires very small amount of time to guide PSOGSA and GPS to a better solution. For function 13 onwards, the time required by MPSOGSA or MGPS almost coincides with that of PSOGSA and GPS respectively. For functions 1–12, there is a small increase in the time requirement but considering the improvement in the results, this amount of time increase is quite insignificant.

5. Applications of the proposed algorithms to solve Engineering Design Problems

Engineering design involves building and designing of products and/or processes. It is a decision-making process requiring complex objective function optimization. Meta-heuristic methods (Simulated Annealing or Tabu search (Hussin and Stützle, 2014)) serve as a better approach than traditional optimization methods like random walk, exhaustive search or steepest descent method. Meta-heuristics converge to an optimal solution and can handle non-convex and non-differentiable functions. Engineering design problems have large number of variables, whereas their influence on the objective function can be very complicated. Therefore, in this paper, five classical engineering design problems *viz.* spring, gear train, welded beam, pressure vessel and closed coil helical spring design have been considered which are discussed in Sections 5.1, 5.2, 5.3, 5.4 and 5.5 respectively. These problems contain various local optima, whereas only global optimum is required. Hence, there is a need for effective and efficient optimization methods for them. In this section, various experiments on these benchmark problems are reported to verify the performance of the proposed algorithms. All the experiments are performed over 30 independent runs for 1000 iterations. The constraints for the problems can be found in Kohli and Arora (2017). The results for each of the functions have been shown in Table 11 and they have been compared with the hybrid algorithms, GPS and PSOGSA. The respective variable

Table 7
Comparison of results of optimization algorithms on multimodal functions of fixed dimension. The best, average and standard deviation of different algorithms are provided.

Function	Value heads	GSA	PSO	GPS	PSOGSA	LCA	VPL	SLC	**Proposed algorithms**	
									MGPS	MPSOGSA
F_{14}	Best	1.0E+00	**1.0E+00**	1.0E+00	1.0E+00	9.98E−01	9.98E−01	1.06E−04	**1.0E+00**	1.0E+00
	Avg.	3.8E+00	**1.0E+00**	1.0E+00	1.39E+00	3.33E−16	9.98E−01	0.00E+00	**1.0E+00**	1.51E+00
	Sd.	2.6E+00	**3.2E−17**	5.8E−01	8.75E−01	9.98E−01	2.32E−13	9.86E−32	**2.84E−17**	6.81E−01
F_{15}	Best	1.4E−03	3.1E−04	**3.1E−04**	3.1E−04	9.95E−04	2.45E−05	0.00E+00	3.53E−04	**3.23E−04**
	Avg.	4.1E−03	1.2E−03	**4.1E−04**	7.16E−04	4.84E−04	1.25E−03	2.22E−03	9.05E−04	**3.59E−04**
	Sd.	3.2E−03	4.0E−03	**3.4E−04**	2.19E−04	1.29E−03	3.08E−04	6.66E−16	4.43E−14	**4.07E−05**
F_{16}	Best	−1.0E+00	−1.0E+00	−1.0E+00	−1.3E+00	−1.01E+00	−1.03E+00	2.92E−04	**−1.03E+00**	**−1.03E+00**
	Avg.	−1.0E+00	−1.0E+00	−1.0E+00	−1.3E+00	3.26E−01	−1.03E+00	4.49E−04	**−1.03E+00**	**−1.03E+00**
	Sd.	4.0E−16	2.3E−16	2.8E−16	2.28E−16	4.50E−01	2.56E−06	1.14E−06	**0.00E+00**	**6.83E−17**
F_{17}	Best	4.0E−01	4.0E−01	4.0E−01	**3.98E−01**	3.98E−01	3.98E−01	0.00E+00	3.98E−01	**3.98E−01**
	Avg.	4.0E−01	4.0E−01	4.0E−01	**3.98E−01**	3.98E−01	3.98E−01	1.78E−15	3.98E−01	**3.98E−01**
	Sd.	3.4E−16	3.4E−16	3.4E−16	**0.00E+00**	1.11E−16	2.69E−06	0.00E+00	1.49E−05	**0.00E+00**
F_{18}	Best	3.0E+00	3.0E+00	3.0E+00	**3.00E+00**	3.00E+00	3.00E+00	1.93E−14	**3.00E+00**	3.00E+00
	Avg.	3.0E+00	3.0E+00	3.0E+00	**3.00E+00**	3.00E+00	3.00E+00	4.81E−82	**3.00E+00**	3.00E+00
	Sd.	2.2E−15	3.1E−15	1.6E−15	**8.40E−16**	9.33E−07	7.58E−05	2.8E−152	**6.09E−16**	9.38E−15
F_{19}	Best	−3.9E+00	−3.9E+00	−3.9E+00	**−3.86E+00**	−1.96E−01	−3.85E+00	−4.10E−77	**−3.85E+00**	−3.88E+00
	Avg.	−3.6E+00	−3.9E+00	−3.9E+00	**−3.86E+00**	4.96E−02	−3.77E+00	−1.36E+01	**−3.87E+00**	−3.88E+00
	Sd.	3.0E−01	3.1E−15	3.1E−15	**2.19E−15**	2.81E−02	9.37EE-02	1.27E−04	**2.70E−02**	1.63E−16
F_{20}	Best	−3.3E+00	−3.3E+00	−3.31E+00	**−3.32E+00**	−3.00E+00	−3.32E+00	−8.60E+03	−3.32E+00	**−3.32E+00**
	Avg.	−1.9E+00	−3.3E+00	−3.3E+00	**−3.31E+00**	5.89E−01	−3.28E+00	8.88E−16	−3.26E+00	**−3.32E+00**
	Sd.	5.4E−01	5.5E−02	2.4E−02	**6.07E−01**	1.54E+00	5.41E−02	0.00E+00	4.75E−02	**0.00E+00**
F_{21}	Best	−5.1E+00	−1.0E+01	**−1.0E+01**	−1.02E+01	−3.40E+00	**−1.02E+01**	3.00E−25	−4.99E+00	−1.02E+01
	Avg.	−5.1E+00	−7.2E+00	**−8.5E+00**	−6.17E+00	5.67E−01	**−9.30E+00**	9.97E−01	−3.61+00	−7.90E+00
	Sd.	7.4E−03	3.3E+00	**3.1E+00**	3.74E+00	5.23E−01	**1.90E+00**	8.89E−04	9.93E−01	2.87E+00
F_{22}	Best	−1.0E+01	**−1.0E+01**	−1.0E+01	−1.04E+01	−2.09E+00	−1.04E+01	−1.03E+00	−4.83E+00	**−1.04E+01**
	Avg.	−7.5E+00	**−9.1E+00**	−1.0E+01	−8.87E+00	3.63E−01	−8.99E+00	3.00E+00	−3.76+00	**−1.04E+01**
	Sd.	2.7E+00	**2.8E+00**	7.2E−15	3.14E+00	7.00E−01	2.35E+00	3.00E−01	1.22E+00	**2.40E−05**
F_{23}	Best	−1.1E+01	−1.1E+01	−1.1E+01	−1.05E+01	−2.06E+00	−1.05E+00	**−3.27E+00**	−4.81E+00	**−1.05E+01**
	Avg.	−1.0E+01	−9.4E+00	−1.0E+01	−7.9E+00	4.22E−01	−9.40E+00	**−1.04E+01**	−3.95E+00	**−1.05E+01**
	Sd.	7.8E−01	2.8E+00	1.6E+00	3.69E+00	9.31E−01	2.28E+00	**1.05E+01**	9.55E−01	**4.76E−06**

Table 8
Comparing best, average and standard deviation of proposed algorithms for unimodal functions with OBSCA, SCA, OBPSO, SSO.

Function	Value heads	**Proposed algorithms**		OBSCA	SCA	OBPSO	SSO
		MGPS	MPSOGSA				
F_1	Best	**4.12E−21**	3.10E−13	**1.75E−75**	6.89E−01	1.85E−07	1.65E−01
	Avg.	**9.38E−19**	1.92E−09	**1.82E−74**	5.43E+00	2.86E−06	1.90E−01
	Sd.	**1.57E−18**	3.67E−09	**2.90E−11**	1.54E+01	1.23E−05	8.47E−17
F_2	Best	**1.40E−20**	2.11E−14	**3.84E−45**	1.25E−02	6.27E−03	1.00E+00
	Avg.	**1.40E−20**	1.09E−10	**1.09E−42**	2.37E−02	5.69E−02	2.05E+00
	Sd.	**2.21E−20**	4.48E−10	**2.90E−11**	4.35E−02	6.95E−02	9.03E−16
F_3	Best	**1.17E−06**	2.5E−02	2.00E+00	3.65E+03	5.45E+00	1.12E+02
	Avg.	**5.57E−07**	9.98E−01	2.05E+01	1.02E+04	6.36E+01	1.14E+02
	Sd.	**5.52E−06**	7.1E+01	2.64E+00	6.38E+02	6.76E+01	2.89E−14
F_4	Best	**1.17E−06**	8.30E−05	**4.5E−34**	2.65E+00	1.89E+00	1.75E+00
	Avg.	**5.52E−06**	3.19E−04	**3.22E−32**	3.63E+01	2.18E+00	2.09E+00
	Sd.	**1.88E−06**	2.74E−04	**1.19E−01**	1.38E+01	1.03E+00	9.03E−16
F_5	Best	**4.19E+00**	**2.24E+01**	1.87E+00	5.54E+02	3.56E+01	6.65E+00
	Avg.	**2.45E+01**	**2.51E+01**	2.82E+01	6.30E+04	5.17E+01	4.54E+01
	Sd.	**9.47E+00**	**6.65E−01**	1.80E−01	1.98E+05	3.04E+01	1.45E−14
F_6	Best	4.79E−06	**1.75E−05**	3.75E+00	2.78E+02	**1.05E−08**	1.45E−01
	Avg.	2.21E−02	**1.26E−02**	4.70E+00	6.30E+04	**1.16E−06**	1.91E−01
	Sd.	2.39E−02	**5.59E−02**	3.32E−01	1.49E+01	**3.92E−06**	8.47E−17
F_7	Best	8.51E−04	**1.13E−04**	**1.25E−05**	1.08E−01	2.35E−03	2.78E−01
	Avg.	1.73E−02	**1.6E−02**	**2.13E−04**	1.35E−01	2.92E−02	3.82E−01
	Sd.	1.01E−02	**1.0E−02**	**2.87E−03**	1.60E−01	1.60E−02	2.26E−16

values for each of the functions are given in Table 12. From the comparison of the results obtained over benchmark functions presented in Tables 5–10, we can see that VPL and LCA are two algorithms which have performed quite well in the scenario. That is why we have also applied them over engineering design problems and compared their results with MBPSOGSA and MGPS in Table 11. Following are specific engineering problems which can be solved using the proposed algorithms efficiently.

5.1. Tension/compression spring design problem

The weight of the spring is based on three decision variables, *namely*, the wire diameter (d), mean coil diameter (D) and the number of active coils (N). The weight is minimized subjected to three inequality constraints and the objective function in Eq. (24). A population particle is given as, $\vec{x} = [x_1 x_2 x_3] = [d\,D\,N]$.

$$EF_1 = (x_3 + 2)\, x_2 x_1^2 \tag{24}$$

Table 9
Comparing best, average and standard deviation of proposed algorithms for multimodal functions with OBSCA, SCA, OBPSO, SSO.

Function	Value heads	**Proposed algorithms**		OBSCA	SCA	OBPSO	SSO
		MGPS	MPSOGSA				
F_8	Best	**−9.11E+03**	−7.86E+03	−6.85E+03	−7.00E+03	−7.24E+03	**−9.89E+03**
	Avg.	**−7.46E+03**	−7.31E+03	−3.53E+03	−3.70E+03	−6.06E+03	**−8.90E+03**
	Sd.	**6.21E+02**	4.76E+02	2.74E+02	3.03E+02	1.00E+03	**5.55E−12**
F_9	Best	1.14E−13	**1.29E−11**	**0.00E+00**	3.45E+01	2.45E+00	6.75E+01
	Avg.	3.47E+00	**2.89E−07**	**0.00E+00**	4.90E+01	4.58E+01	7.53E+01
	Sd.	2.84E+00	**1.06E−06**	**2.08E−09**	4.06E+01	1.35E+01	1.45E−14
F_{10}	Best	**2.86E−11**	5.62E−07	**3.84E−13**	4.56E+00	1.15E+00	3.65E−01
	Avg.	**1.42E−08**	5.83E−05	**8.88E−16**	1.66E+01	1.52E+00	4.85E−01
	Sd.	**3.59E−08**	5.28E−05	**2.07E+00**	7.12E+00	8.02E−01	3.95E−16
F_{11}	Best	**0.0E+00**	**6.79E−14**	2.25E−02	7.75E−02	3.26E−03	1.04E−02
	Avg.	**8.49E−05**	**1.09E−11**	1.00E−01	8.88E−01	2.69E−02	1.04E−02
	Sd.	**2.96E−04**	**1.75E−11**	7.00E−02	3.13E−01	3.84E−02	0.00E+00
F_{12}	Best	**1.00E−03**	**2.66E−08**	4.73E−01	1.45E+04	1.54E−02	2.56E+00
	Avg.	**1.9E−03**	**1.2E−02**	5.72E−01	2.89E+04	1.56E−01	3.27E+00
	Sd.	**5.32E−04**	**2.4E−03**	1.80E−01	1.07E+05	2.85E−01	4.52E−16
F_{13}	Best	**2.56E−02**	5.93E−01	1.75E+00	7.8E+03	3.45E−02	1.14E+01
	Avg.	**4.33E−01**	9.73E−01	2.41E+00	6.75E+04	7.83E−02	1.14E−01
	Sd.	**3.1E−01**	1.88E−01	1.69E−01	1.98E+05	2.00E−01	**0.00E+00**

Table 10
Comparing best, average and standard deviation of proposed algorithms for multimodal functions with a fixed dimension, with OBSCA, SCA, OBPSO, SSO.

Function	Value heads	**Proposed algorithms**		OBSCA	SCA	OBPSO	SSO
		MGPS	MPSOGSA				
F_{14}	Best	**1.0E+00**	**1.0E+00**	1.37E+00	2.09E+00	1.07E+00	1.45E+00
	Avg.	**1.0E+00**	**1.51E+00**	2.64E+00	2.18E+00	3.40E+00	2.98E+00
	Sd.	**2.84E−11**	**6.81E−01**	3.11E+00	2.49E+00	2.74E+00	4.52E−16
F_{15}	Best	3.53E−04	**3.23E−04**	4.56E−04	2.89E−01	2.34E−04	**3.45E−04**
	Avg.	9.05E−04	**3.59E−04**	6.58E−04	1.08E+00	1.88E−03	**7.45E−04**
	Sd.	4.43E−14	**4.07E−05**	2.83E−04	3.78E−04	5.04E−03	**3.31E−19**
F_{16}	Best	**−1.03E+00**	**−1.03E+00**	−1.03E+00	−1.03E+00	−1.03E+00	−1.03E+00
	Avg.	**−1.03E+00**	**−1.03E+00**	−1.05E+00	−1.01E+00	−1.02E+00	−1.04E+00
	Sd.	**0.00E+00**	**6.83E−17**	8.51E−06	4.46E−05	6.45E−16	9.06E−16
F_{17}	Best	**3.98E−01**	**3.98E−01**	3.98E−01	3.99E−01	**3.98E−01**	**3.98E−01**
	Avg.	**3.98E−01**	**3.98E−01**	3.99E−01	4.00E−01	**3.98E−01**	**3.98E−01**
	Sd.	**1.49E−05**	**0.00E+00**	6.55E−04	1.43E−03	**0.00E+00**	**0.00E+00**
F_{18}	Best	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**
	Avg.	**3.00E+00**	3.01E+00	3.10E+00	3.13E+00	3.24E+00	**3.00E+00**
	Sd.	**6.09E−16**	9.38E−15	6.54E−05	1.56E−04	1.22E−15	**0.00E+00**
F_{19}	Best	**−3.85E+00**	**−3.88E+00**	−3.81E+01	−3.56E−01	−3.45E−01	−2.86E−01
	Avg.	**−3.87E+00**	**−3.88E+00**	−3.00E−01	−3.00E−01	−3.00E−01	−2.86E−01
	Sd.	**2.70E−02**	**1.63E−16**	2.26E−16	2.26E−16	2.26E−16	0.00E+00
F_{20}	Best	−3.32E+00	**−3.32E+00**	−3.25E+00	−3.20E+00	−3.29E+00	**−3.31E+00**
	Avg.	−3.26E+00	**−3.32E+00**	−3.10E+00	−3.04E+00	−3.29E+00	**−3.31E+00**
	Sd.	4.75E−02	**0.00E+00**	3.94E−02	1.16E−01	5.35E−02	**0.00E+00**
F_{21}	Best	−4.99E+00	−1.02E+01	**−1.04E+01**	−6.57E+01	−7.65E+01	**−1.05E+01**
	Avg.	−3.61+00	−7.90E+00	**−9.06E+00**	−2.20E+00	−6.24E+00	**−9.49E+00**
	Sd.	9.93E−01	2.87E+00	**1.76E+00**	1.71E+00	3.74E+00	**3.61E−15**
F_{22}	Best	−4.83E+00	**−1.04E+01**	−9.95E+00	−6.78E+00	−9.95E+00	**−1.04E+01**
	Avg.	−3.76+00	**−1.04E+01**	−9.93E+00	−4.27E+00	−8.33E+00	**−1.04E+01**
	Sd.	1.22E+00	**2.40E−05**	2.73E−01	1.43E+00	3.26E+00	**5.42E−15**
F_{23}	Best	−4.81E+00	**−1.05E+01**	−1.02E+01	−6.67E+00	−9.87E+00	**−1.05E+01**
	Avg.	−3.95E+00	**−1.05E+01**	−1.01E+01	−3.34E+00	−8.21E+00	**−1.05E+01**
	Sd.	9.55E−01	**4.76E−06**	2.56E−01	1.78E+00	3.41E+00	**0.00E+00**

5.2. Gear train design problem

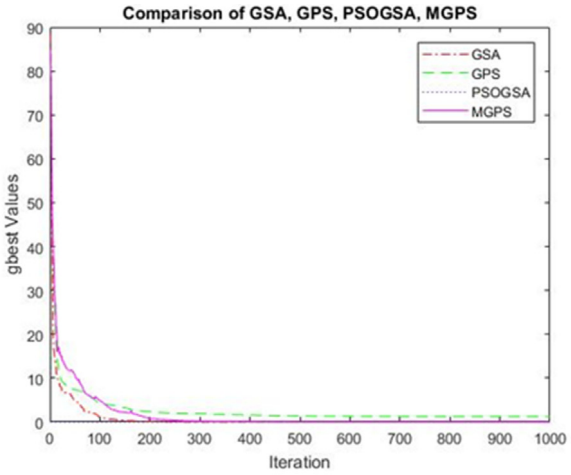
Here the cost of gear ratio, of the gear train, is minimized. The problem has no equality or inequality constraint except a boundary constraint. It consists of four decision variables $n_A(x_1)$, $n_B(x_2)$, $n_D(x_3)$, $n_F(x_4)$ using which the gear ratio can be formulated as $n_B n_D / n_F n_A$. The objective function to be minimized is,

$$EF_2(x) = \left((1/6.931) - \left(x_3 x_2 / x_1 x_4\right)\right)^2$$
$$\text{Subject to, } 12 \le x_i \le 60 \tag{25}$$

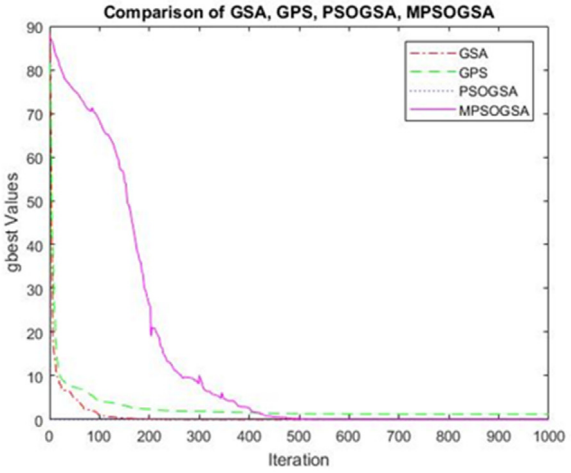
5.3. Welded beam design problem

This is a minimization problem having four variables namely weld thickness (h), length of the bar attached to the weld (l), bar's height (t), and bar's thickness (b). The constraints for this problem include bending stress (θ), bean deflection (δ), shear stress (τ), buckling load (P_c) and other constraints. The population point is taken as $\vec{x} = \left[x_1 x_2 x_3 x_4\right] = [hltb]$. The objective function is,

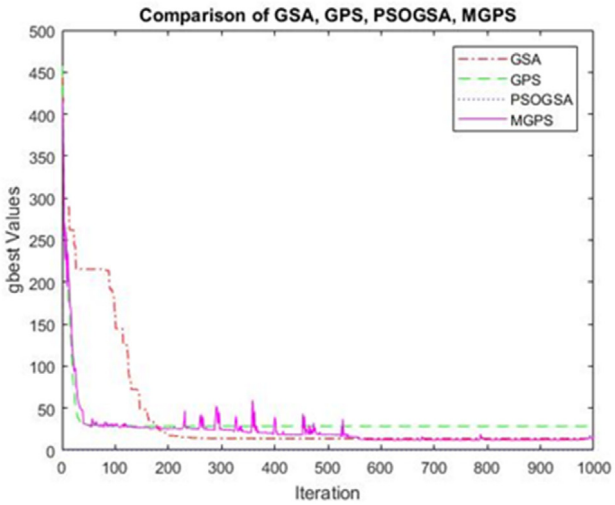
$$EF_3(x) = 1.1047 x_1^2 x_2 + 0.04811 x_3 x_4 \left(14.0 + x_2\right) \tag{26}$$



A: Graph comparing $gbest$ of function F_4 at each iteration using GSA, GPS, PSOGSA and MGPS algorithms

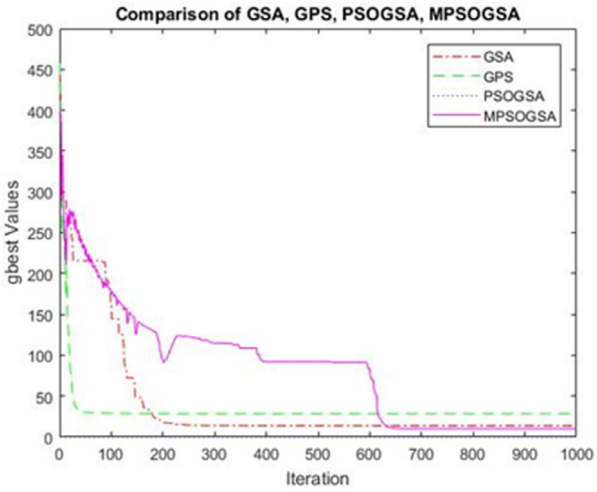


B: Graph comparing $gbest$ of function F_4 at each iteration using GSA, GPS, PSOGSA and MPSOGSA algorithms

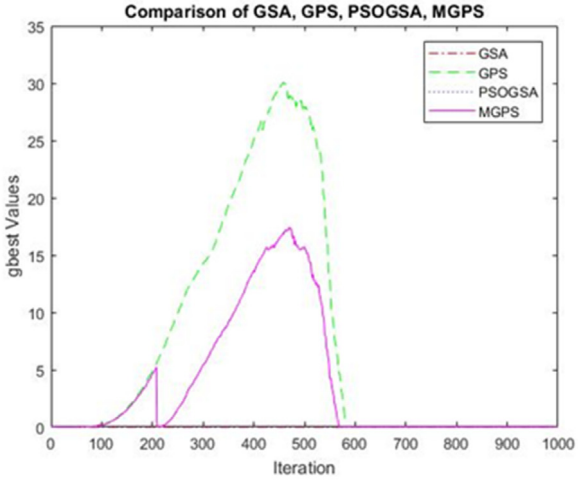


C: Graph comparing $gbest$ of function F_9 at each iteration using GSA, GPS, PSOGSA and MGPS algorithms

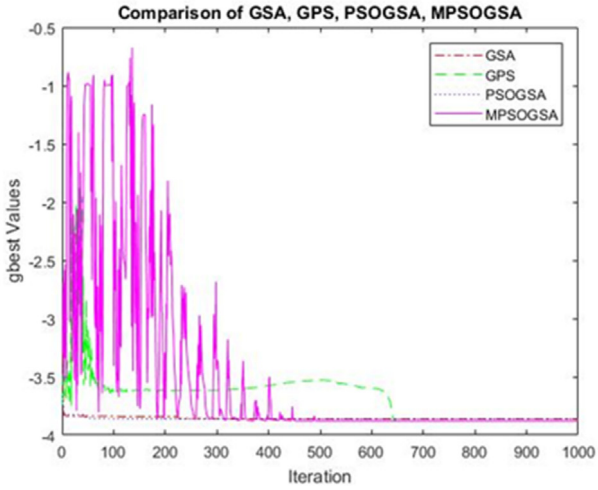
Fig. 3. (A–F) Variations of the value of *gbest* of the particles in the population versus iterations for MGPS/MPSOGSA, along with their ancestors (GPS and PSOGSA), PSO and GSA. A, C, E — contain the plots of the MGPS algorithm and B, D, F contain the plots of MPSOGSA algorithm.



D: Graph comparing $gbest$ function F_9 at each iteration using GSA, GPS, PSOGSA and MPSOGSA algorithms

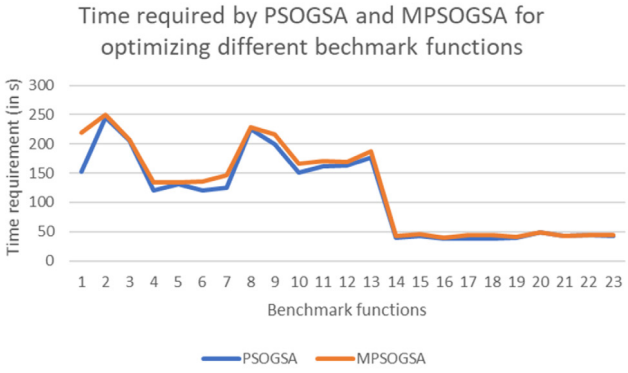


E: Graph comparing $gbest$ function F_{15} at each iteration using GSA, GPS, PSOGSA and MGPS algorithms

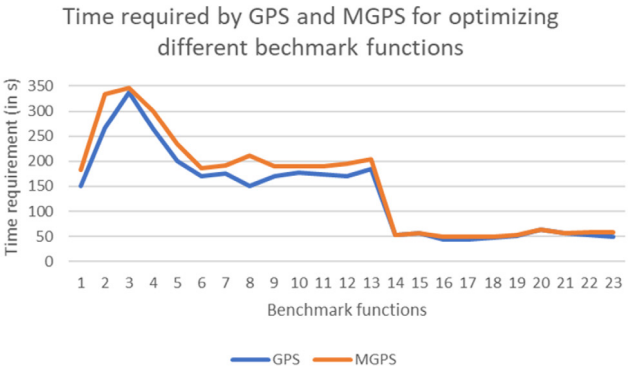


F: Graph comparing $gbest$ of function F_{19} at each iteration using GSA, GPS, PSOGSA and MPSOGSA algorithms

Fig. 3. (*continued*).



A: Graph representing time requirement against 23 benchmark functions for PSOGSA and MPSOGSA.



B: Graph representing time requirement against 23 benchmark functions for GPS and MGPS.

Fig. 4. (A–B) The time required by PSOGSA and GPS before and after addition of mutation operation for optimizing benchmark functions.

Table 11
Function values for the design problems — spring, gear train, welded beam, pressure vessel and closed coil helical spring design problem.

Function	Value heads	GPS	PSOGSA	VPL	LCA	MGPS	MPSOGSA
EF_1	Best	6.9E−03	2.5E−03	1.24E−3	1.26E−3	**2.5E−03**	**2.5E−03**
	Avg.	1.07E−01	2.5E−03	2.37E−2	2.21E−2	**2.5E−03**	**2.5E−03**
	Sd.	2.5E−03	8.64E−10	1.8E−03	3.88E−3	**8.90E−19**	**8.90E−19**
EF_2	Best	2.88E−07	3.6E−03	2.8E−12	2.5E−11	**5.01E−14**	**0.00E+00**
	Avg.	2.00E−03	2.00E−03	2.5E−09	3.8E−08	**8.34E−04**	**0.00E+00**
	Sd.	4.0E−03	4.93E−09	3.9E−06	1.1E−09	**1.80E−03**	**0.00E+00**
EF_3	Best	**0.00E+00**	**0.00E+00**	2.26E+0	1.72E+0	**0.00E+00**	**0.00E+00**
	Avg.	**0.00E+00**	**0.00E+00**	3.21E+0	1.72E+0	**0.00E+00**	**0.00E+00**
	Sd.	**0.00E+00**	**0.00E+00**	4.7E−16	7.1E−15	**0.00E+00**	**0.00E+00**
EF_4	Best	0.00E+00	0.00E+00	6.04E+3	6.06E+3	**0.00E+00**	**0.00E+00**
	Avg.	8.88E+03	3.33E+02	6.87E+3	6.07E+3	**0.00E+00**	**0.00E+00**
	Sd.	4.48E+04	4.79E+02	1.32E+1	11.4E+0	**0.00E+00**	**0.00E+00**
EF_5	Best	**13.75E+00**	**13.75E+00**	40.1E+0	42.8E+0	**13.75E+00**	**13.75E+00**
	Avg.	**13.75E+00**	**13.75E+00**	41.9E+0	43.7E+0	**13.75E+00**	**13.75E+00**
	Sd.	**0.00E+00**	**0.00E+00**	2.3E+00	1.7E+00	**0.00E+00**	**0.00E+00**

5.4. Pressure design vessel problem

Pressure vessel design problem involves minimization of the welding, manufacturing and material cost of the pressure vessel. There are four decision variables involved in this problem which are the thickness of shell (T_S), the thickness of head (T_h) which are discrete decision variables, inner radius (R) and length of the cylindrical section of the vessel (L) which are continuous decision variables. The population point is taken as $\vec{x} = [x_1 x_2 x_3 x_4] = [T_s T_h R L]$. The objective function is,

$$EF_4(\vec{x}) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3 \quad (27)$$

5.5. Closed coil helical spring design problem

The volume of the closed coil helical spring is minimized. Helical spring is made up of closed coil wire having the shape of a helix and is intended for the tensile and compressive load. The population point

Table 12
Value of the parameters after optimization.

Function	Algorithm	x_1	x_2	x_3	x_4	$f(x)$
EF_1	MGPS	0.05	0.25	2.00	N/A	2.5E−03
	MPSOGSA	0.05	0.25	2.00	N/A	2.5E−03
EF_2	MGPS	60	12	43.2837	60	0
	MPSOGSA	29.2062	12	12.0749	34.3865	0
EF_3	MGPS	0.1	0.1	0.1	0.1	0
	MPSOGSA	0.1	0.1	0.1	0.1	0
EF_4	MGPS	0	0	82.7991	10.6423	0
	MPSOGSA	0	0	79.0777	10	0
EF_5	MGPS	0.508	1.27	15	N/A	13.74738
	MPSOGSA	0.508	1.27	15	N/A	13.74738

Note: In case of EF_4 in 5.4 since the lower boundary for both x_1 and x_2 is 0 the algorithm converges towards 0.

is given as $\vec{x} = [x_1 x_2 x_3] = [d\, D N_c]$. There are chiefly two decision variables to consider namely coil diameter(D) and wire diameter(d). The number of coils (N_c) can be fixed beforehand. The volume of the helical spring (U) is given as the minimization function,

$$EF_5 = \frac{\pi^2}{4}\left(N_c + 2\right) D d^2 \tag{28}$$

6. Conclusion and future work

The hybrids of GSA and PSO — GPS and PSOGSA are found to be efficient in single-objective optimization but suffer from premature convergence. This problem is addressed in the present work by the use of mutation and hence better optimizations results are obtained. We have proposed a model of fuzzy mutation based on the distances between the points from the centroid and the population history, which helps the proposed algorithms outperform their ancestors GPS and PSOGSA. The evaluation of the models on benchmark functions provides impressive results. To show the practical application of our proposed algorithms, they have been evaluated on five classic engineering design problems. The results are quite promising and our algorithms outperform their ancestors in most cases or are shoulder to shoulder. This model of mutation is not algorithm-specific and can be applied to any algorithm which suffers from premature convergence like Whale optimization or Harmony search algorithm. Future scope of this work might involve the use of a local and a global change counters to perform mutation.

CRediT authorship contribution statement

Devroop Kar: Methodology, Conceptualization, Software, Formal analysis, Writing - original draft. **Manosij Ghosh:** Methodology, Conceptualization, Software, Writing - original draft. **Ritam Guha:** Software, Formal analysis, Writing - original draft. **Ram Sarkar:** Conceptualization, Writing - review & editing, Supervision. **Laura Garcia-Hernandez:** Writing - review & editing, Supervision. **Ajith Abraham:** Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has not received any funds.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abd Elaziz, M., Oliva, D., Xiong, S., 2017. An improved opposition-based sine cosine algorithm for global optimization. Expert Syst. Appl. 90, 484–500.
Angeline, P.J., 1998. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: International Conference on Evolutionary Programming, pp. 601–610.
Chen, J.E., Otto, K.N., 1995. Constructing membership functions using interpolation and measurement theory. Fuzzy Sets and Systems 73 (3), 313–327.
Chuang, L.-Y., Chang, H.-W., Tu, C.-J., Yang, C.-H., 2008. Improved binary PSO for feature selection using gene expression data. Comput. Biol. Chem. 32 (1), 29–38.
Cuevas, E., Cienfuegos, M., Zaldívar, D., Pérez-cisneros, M., A swarm optimization algorithm inspired in the behavior of the social-spider, 40 (16) (2013) 6374–6384.
Czekalski, P., 2006. Evolution-fuzzy rule based system with parameterized consequences. Int. J. Appl. Math. Comput. Sci. 16, 373–385.
Dasgupta, D., Michalewicz, Z., 2013. Evolutionary Algorithms in Engineering Applications. Springer Science & Business Media.
Deb, K., 1999. Evolutionary algorithms for multi-criterion optimization in engineering design. Evol. Algorithms Eng. Comput. Sci. 2, 135–161.
Dhargupta, S., Ghosh, M., Mirjalili, S., Sarkar, R., 2020. Selective opposition based grey wolf optimization. Expert Syst. Appl. 113389.
Duman, S., Sönmez, Y., Güvenç, U., Yörükeren, N., 2012. Optimal reactive power dispatch using a gravitational search algorithm. IET Gener. Transm. Distrib. 6 (6), 563–576.
Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Micro Machine and Human Science, 1995. MHS'95. Proceedings of the Sixth International Symposium on, pp. 39–43.
El-Zonkoly, A.M., Khalil, A.A., Ahmied, N.M., 2009. Optimal tunning of lead–lag and fuzzy logic power system stabilizers using particle swarm optimization. Expert Syst. Appl. 36 (2), 2097–2106.
Feng, D., Wenkang, S., Liangzhou, C., Yong, D., Zhenfu, Z., 2005. Infrared image segmentation with 2-d maximum entropy method based on particle swarm optimization (PSO). Pattern Recognit. Lett. 26 (5), 597–603.
Gandomi, A.H., Yang, X.-S., Alavi, A.H., 2013. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. 29 (1), 17–35.
Hussin, M.S., Stützle, T., 2014. Tabu search vs. simulated annealing as a function of the size of quadratic assignment problem instances. Comput. Oper. Res. 43, 286–291.
Jeong, Y.-W., Park, J.-B., Jang, S.-H., Lee, K.Y., 2010. A new quantum-inspired binary PSO: application to unit commitment problems for power systems. IEEE Trans. Power Syst. 25 (3), 1486–1495.
Kashan, A.H., 2009. League championship algorithm : A new algorithm for numerical function optimization.
Kohli, M., Arora, S., 2017. Chaotic grey wolf optimization algorithm for constrained optimization problems. J. Comput. Des. Eng..
Li, C., Zhou, J., Xiao, J., Xiao, H., 2012. Parameters identification of chaotic system by chaotic gravitational search algorithm. Chaos Solitons Fractals 45 (4), 539–547.
Lim, Y.W., Lee, S.U., 1990. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. Pattern Recognit. 23 (9), 935–952.
Liu, J., Lampinen, J., 2005. A fuzzy adaptive differential evolution algorithm. Soft Comput. 9 (6), 448–462.
Mirjalili, S., 2016. SCA: a sine cosine algorithm for solving optimization problems. Knowl.-Based Syst. 96, 120–133.
Mirjalili, S., Hashim, S.Z.M., 2010. A new hybrid PSOGSA algorithm for function optimization. In: Proc. ICCIA 2010-2010 Int. Conf. Comput. Inf. Appl., no. 1, pp. 374–377.
Mitaim, S., Kosko, B., 1996. What is the best shape for a fuzzy set in function approximation? In: Proceedings of IEEE 5th International Fuzzy Systems, vol. 2, pp. 1237–1243.
Moghdani, R., Salimifard, K., 2018. Volleyball premier league algorithm. Appl. Soft Comput. J. 64, 161–185.
Moosavian, N., Roodsari, B.K., 2014. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. Swarm Evol. Comput. 17, 14–24.
Omran, M.G.H., Salman, A., Engelbrecht, A.P., 2006. Dynamic clustering using particle swarm optimization with application in image segmentation. Pattern Anal. Appl. 8 (4), 332.
Papa, J.P., Pagnin, A., Schellini, S.A., Spadotto, A., Guido, R.C., Ponti, M., Chiachia, G., Falcao, A.X., 2011. Feature Selection Through Gravitational Search Algorithm, Department of Computing University of São Paulo University of Campinas Institute of Computing. Sort (1), 2052–2055.
Puranik, P., Bajaj, P., Abraham, A., Palsodkar, P., Deshmukh, A., 2009. Human perception-based color image segmentation using comprehensive learning particle swarm optimization. In: Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on, pp. 630–635.
Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., 2009. GSA: A gravitational search algorithm. Inf. Sci. (Ny). 179 (13), 2232–2248.
Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2010. BGSA: Binary gravitational search algorithm. Nat. Comput. 9 (3), 727–745.
Robinson, J., Rahmat-Samii, Y., 2004. Particle swarm optimization in electromagnetics. IEEE Trans. Antennas Propag. 52 (2), 397–407.

Ross, T.J., 2005. Fuzzy Logic with Engineering Applications. John Wiley & Sons.
Rutkowska, A., 2016. Influence of membership function's shape on portfolio optimization results. J. Artif. Intell. Soft Comput. Res. 6 (1), 45–54.
Sadollah, A., 2018. Introductory chapter: which membership function is appropriate in fuzzy system? In: Fuzzy Logic Based in Optimization Methods and Control Systems and Its Applications. IntechOpen.
Schaefer, G., Závišek, M., Nakashima, T., 2009. Thermography based breast cancer analysis using statistical features and fuzzy classification. Pattern Recognit. 42 (6), 1133–1137.
Schott, J.R., 1995. Fault Tolerant Design using Single and Multicriteria Genetic Algorithm Optimization. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH.
Soleimanpour-Moghadam, M., Nezamabadi-Pour, H., Farsangi, M.M., 2014. A quantum inspired gravitational search algorithm for numerical function optimization. Inf. Sci. (Ny). 267, 83–100.
Sun, G., Zhang, A., 2013. A hybrid genetic algorithm and gravitational search algorithm for image segmentation using multilevel thresholding. In: Iberian Conference on Pattern Recognition and Image Analysis, pp. 707–714.
Tong, C., 2014. Gravitational search algorithm based on simulated annealing. J. Converg. Inf. Technol. 9 (2), 231.
Tsai, H.C., Tyan, Y.Y., Wu, Y.W., Lin, Y.H., 2013. Gravitational particle swarm. Appl. Math. Comput. 219 (17), 9106–9117.
Van Der Merwe, D.W., Engelbrecht, A.P., 2003. Data clustering using particle swarm optimization. In: 2003 Congr. Evol. Comput. 2003 CEC 03, vol. 1, pp. 215–220.
Wang, H., Li, H., Liu, Y., Li, C., Zeng, S., 2007. Opposition-based particle swarm algorithm with Cauchy mutation. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 4750–4756.
Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1 (1), 67–82.
Wu, D., 2012. Twelve considerations in choosing between Gaussian and trapezoidal membership functions in interval type-2 fuzzy logic controllers. In: 2012 IEEE International Conference on Fuzzy Systems, pp. 1–8.
Xue, B., Zhang, M., Member, S., Browne, W.N., 2012. Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Trans. Cybern. 1–16.
Yin, M., Hu, Y., Yang, F., Li, X., Gu, W., 2011. A novel hybrid k-harmonic means and gravitational search algorithm approach for clustering. Expert Syst. Appl. 38 (8), 9319–9324.
Zadeh, L.A., 1996. Fuzzy sets. In: Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers By Lotfi A Zadeh. World Scientific, pp. 394–432.
Zhan, Z.-H., Zhang, J., Li, Y., Chung, H.S.-H., 2009. Adaptive particle swarm optimization. IEEE Trans. Syst. Man Cybern. B 39 (6), 1362–1381.
Zhang, W., Niu, P., Li, G., Li, P., 2013. Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm. Knowl.-Based Syst. 39, 34–44.