Expert Systems With Applications 151 (2020) 113389

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Selective Opposition based Grey Wolf Optimization

Souvik Dhargupta[a,*], Manosij Ghosh[a], Seyedali Mirjalili[b], Ram Sarkar[a]

[a] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
[b] Centre for Artificial Intelligence Research and Optimisation, Torrens University Australia, 90 Bowen Terrace, Fortitude Valley, Queensland 4006, Australia

ARTICLE INFO

Article history:
Received 13 October 2019
Revised 13 March 2020
Accepted 14 March 2020
Available online 17 March 2020

Keywords:
Grey Wolf Optimizer
Opposition-based Learning
Spearman's coefficient
Selective opposition

ABSTRACT

The use of metaheuristics is widespread for optimization in both scientific and industrial problems due to several reasons, including flexibility, simplicity, and robustness. Grey Wolf Optimizer (GWO) is one of the most recent and popular algorithms in this area. In this work, opposition-based learning (OBL) is combined with GWO to enhance its exploratory behavior while maintaining a fast convergence rate. Spearman's correlation coefficient is used to determine the omega (ω) wolves (wolves with the lowest social status in the pack) on which to perform opposition learning. Instead of opposing all the dimensions in the wolf, a few dimensions of the wolf are selected on which opposition is applied. This assists with avoiding unnecessary exploration and achieving a fast convergence without deteriorating the probability of finding optimum solutions. The proposed algorithm is tested on 23 optimization functions. An extensive comparative study demonstrates the superiority of the proposed method. The source code for this algorithm is available at "https://github.com/dhargupta-souvik/sogwo"

1. Introduction

Metaheuristics can be single solution based (Van Laarhoven and A.E., 1987) or population based (Ghosh, Begum, Sarkar, Chakraborty, & Maulik, 2019; Ghosh, Kundu, Ghosh, & Sarkar, 2019). Single solution-based algorithms start with one candidate solution, which is improved on every iteration. Simulated Annealing (SA) (Van Laarhoven and A.E., 1987) is an example for single solution based algorithms. On contrary, population-based algorithms begin with a population of candidate solutions. These solutions gradually evolve through iterations to reach better solutions. Swarm or evolution based algorithms are generally population based.

Metaheuristics again can be categorized into: evolutionary, physics-based, and swarm-based (Mirjalili & Lewis, 2016; Mirjalili, Mirjalili, & Lewis, 2014). Evolutionary algorithm (EA) generally imitates the technique of biological evolution scheme found in nature. Of these algorithms, the most popular one is Genetic Algorithm (GA) (Malakar, Ghosh, Bhowmik, Sarkar, & Nasipuri, 2019), which is based upon Darwinian evolution theory. EA falls under the class of generic population based metaheuristic algorithms (Ghosh, Malakar, Bhowmik, Sarkar, & Nasipuri, 2019; Liu & Lampinen, 2005), which are widely used in field of Artificial Intelligence (AI) to solve optimization problems with high computational complexity. EAs improve the initial solutions by some means following one or more natural or biological process(es). To solve the optimization problem, an EA generally searches the specified domain of a given problem as follows: initially, a random population of individual candidate solutions is selected and a fitness function is used to determine their feasibility. After that, by the repeated application of operations and fitness function evaluation, the candidate solutions evolve towards the global optimum until the maximum iteration is reached or it yields a solution that is sufficiently near to the global optimum.

Physics based algorithms emulate the physical laws of nature. For example, SA is inspired by metal annealing process. Gravitational Search Algorithm (GSA) (Rashedi, Nezamabadi-pour, & Saryazdi, 2009), Ray Optimization Algorithm (ROA) (Kaveh & Khayatazad, 2012) and Harmony Search Algorithm (HSA) (Singh, Das, Sarkar, & Nasipuri, 2018) are based on several physical events such as gravitational interactions, laws of refraction, and harmony among simultaneously played sound waves respectively.

Swarm based algorithms generally imitate the behavior of a creature who lives in groups. Ant Colony Optimization (ACO) is a mimicry of the method of food searching by a colony of ants. Particle Swarm Optimization (PSO) (Kennedy, 2011) is inspired by a flock of birds' behavior. Cuckoo Search (CS) algorithm is inspired by the parasitic breeding process of cuckoo. Artificial Bee Colony (ABC) algorithm is designed following the behavior of bee swarm. Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014) is based on hunting behavior grey wolves.

The advantages of population-based algorithms over single solution-based algorithms are cooperative search, greater explo-

* Corresponding author.
E-mail address: souvikdhargupta@gmail.com (S. Dhargupta).

https://doi.org/10.1016/j.eswa.2020.113389
0957-4174/

ration through sharing of information among the candidate solutions and greater capacity to avoid local optima. Evolutionary techniques change the population at a constant rate, and this changes the equilibrium between exploration and exploitation. Swarm based search techniques, however, are able to get a better equilibrium of exploration and exploitation. Keeping all these in mind, GWO is used for optimization in our proposed work. The adaptive values of parameters in GWO allow it to make exploitation and exploration balanced. All that said, we make a decent attempt to further increment the exploration capacity of GWO through the use of opposition-based learning (OBL). OBL helps in further exploration and better probability to converge faster to the optimum solution. Selection of candidate solutions in each iteration using Spearman's Rank Correlation Coefficient aids in avoiding unnecessary exploration as well as making the convergence faster.

Rest of the paper is divided into following sections. An in-depth literature review (i.e. the recent works done on function optimization) is outlined in Section 2. In Section 3, the prerequisite concepts and theories such as GWO, Spearman coefficient and OBL are explained. The proposed method is discussed in Section 4. The results and discussion about the outcomes of our method can be found in Section 5. In Section 6, we conclude our paper outlining the proposed work and possible further work that can be done.

2. Related work

Scientists have developed many metaheuristic algorithms over the past few years. A large number of works can be found in the literature which have taken inspiration from various natural occurrences such as pack behavior of grey wolves, the movement of celestial objects and so on. A few prominent works are briefed hereafter.

In case of physics based algorithms, generally speaking, the random initial search agents move towards the optimal solutions according to some physical rule via interacting among themselves. GSA is modelled upon the laws of gravity where the heaviest mass object (here, the global optimum) attracts other objects (candidate solutions) in force inversely proportional to distance between the two objects. A modification of GSA is Gravitational Local Search Algorithm (GLSA) (Webster & Bernhard, 2003). GLSA has two versions namely GLSA1, which calculates the gravitational force to solve optimization problem, and GLSA2, which calculates gravitational field to do the same. However, in all these algorithms no repulsion is used i.e. particles only attract each other. In Abedinpourshotorban, Shamsuddin, Beheshti, and Jawawi (2016), Abedinpourshotorban et al. have introduced a metaheuristic called Electromagnetic Field Optimization (EFO), which mimics the behavior of different polarity electromagnets and golden ratio. In this algorithm, all the particles are given a polarity – positive (particles with good fitness), neutral and negative (low fitness particles). The algorithm tries to find a solution by moving towards positive particles and away from negative particles.

In 2016, Muthiah-Nakarajan and Noel (2016) have proposed a metaheuristic named Galactic Swarm Optimization (GSO) which mimics the movement of galaxy and its components such as stars and super clusters. The particles are divided into sub-swarms to emulate galactic motion. Another different physics inspired algorithm is Water-Evaporation Optimization (WEO) (Kaveh & Bakhshpoori, 2016), which is modelled upon the evaporation of water molecules on solid surface of different wettability. In 2017, Baykasoğlu and Ozsoydan (2018) have proposed a new extension of swarm intelligence based metaheuristic called Weighted Superposition Attraction (WSA) algorithm to solve binary dynamic optimization problems. In 2018, Baykasoğlu, Ozsoydan, and Senol (2018) have proposed the application of WSA algorithm to solve binary optimization problems.

PSO is a well-established swarm based algorithm where each particle moves according to the positions of the local optima and global optima. ACO (Aghdam, Ghasem-Aghaee, & Basiri, 2009) mimics the ways ant forage for food by leaving behind a trail of chemicals called pheromone as they travel allowing more ants to follow that trail to reach food. Many swarm algorithms have also been proposed in recent years. Gravitational Particle Swarm (GPS) (Tsai, Tyan, Wu, & Lin, 2013) hybridizes PSO and GSA in such a way that each particle is influenced by the combination of PSO and GSA operations. In another such hybrid called PSOGSA (Mirjalili & Hashim, 2010), the equations of PSO are modified to account for getting stuck in a local optima. In 2016, Zhang et al. (2017) have suggested an algorithm based on decision-making abilities of human nature. The algorithm comprises five phases of decision making which are experience, others, group thinking, leader and inspiration. In this method, at each step of iteration, the best search agent obtained so far updates its position using local random walk. In 2016, Punnathanam and Kotecha (2016) have proposed a novel metaheuristic named Yin-Yang-Pair Optimization (YYPO) based on comparing exploration and exploitation with the Yin-Yang, two complementary but opposite identities. However, as many other optimization algorithms, for rotated unimodal functions this algorithm fails to converge to global optima. In 2018, Shayanfar and Gharehchopogh (2018) have proposed a novel algorithm based on farmland fertility that divides the search space into several parts and optimizes using two types of memories. In 2018, Mortazavi, Toğan, and Nuhoğlu (2018) have proposed an algorithm named Iterative Search Algorithm, combining features of integrated PSO, and Teaching and Learning Based Optimization.

Many swarm algorithms are inspired by animals and their behavior. Jahani and Chizari (2018) have come up with a novel algorithm known as Mouth Breeding Fish (MBF) algorithm, where they have imitated the movement, dispersion and protective behavior of MBF to solve optimization problems. This method has promised better results when applied on complex problems such as functions with increased number of dimensions. In 2016, Ebrahimi and Khamehchi (2016) have proposed Sperm Whale Algorithm (SWA) based on the lifestyle of the said animal. One of the key features of this method is that it applies the best and worst search agents at each iteration to reach the global optima. A unique nature inspired algorithm named Virus Colony Search (VCS) has been proposed by Li, Zhao, Weng, and Han (2016). It is based upon the survival of virus by defusing and infecting into host bodies. In 2018, Pierezan and Coelho (2018) have proposed Coyote Optimization Algorithm (COA) mimicking the behavior of *canis latrans*, maintaining balance between exploration and exploitation. In 2018 Klein and dos Santos Coelho (2018) have proposed an algorithm based on meerkats behavior, using animal behavior for designing algorithm as well as for parameter selection. In 2018, Klein, Mariani, and Coelho (2018) have proposed an algorithm inspired by the behavior of cheetah. In 2019, de Vasconcelos Segundo, Mariani, and dos Santos Coelho (2019) have proposed a novel algorithm based on the hunting behavior of falcons named Falcon Optimization Algorithm (FOA) and applied it to the design of heat exchangers. In 2019, Shadravan, Naji, and Bardsiri (2019) have proposed Sailfish algorithm based on the hunting nature of sailfish using sailfish population for exploitation and sardine population for exploration.

Human biological functions have inspired the researchers to devise a number of algorithms too. In Jaddi, Alvankarian, and Abdullah (2017) a population-based algorithm imitating the glomerular filtration technique of kidney is proposed. This method performs a smaller number of function evaluations. In 2015, Tang et al. have modelled (Tang, Dong, Jiang, Li, & Huang, 2015) a metaheuristic by imitating the invasive growth of tumor by absorbing nutrients from microenvironment. In this method, a new framework with 3 types of cell populations (proliferative, quiescent and dying), four types of cell roles (proliferative, quiescent, dying and invasive) and five types of search rules (growth of all four types of cells and random walk for all of them) have been presented.

EAs have emerged as a very robust and practical search method for global optimum. In the last few years, numerous algorithms have been proposed in the literature. EA generally works through use of three general procedures namely selection, crossover and mutation. Besides GA, Differential Evolution (DE) is another popular approach for solving nonlinear and non-differentiable optimization problems. Though it has good exploration ability, it lacks exploitation capacity. Also, DE requires problem-specific parameter setting. Evolutionary Programming (EP) allows the structure of the optimizing program to be fixed to evolve the parameters. On the other hand, Evolution Strategy (ES) primarily uses mutation and selection as search operators.

Many techniques have been proposed for the improvement of these metaheuristics. Some of these techniques include the use of OBL, chaos, hybridization and so on. The No Free Lunch theorem (Wolpert & Macready, 1997) has kept research in this field alive. Of these techniques, a popular and effective one is OBL (Tizhoosh, 2005). OBL algorithms have attracted the interests of numerous computer scientists in the past years. Many of the previously discussed algorithms such as GSA, SA, GA, PSO, ACO, ABC, DE etc. have been enhanced by the application of OBL on them (Mahdavi, Rahnamayan, & Deb, 2018).

Ergezer and Simon (2014) have introduced fitness-based quasi-reflection concept, where relative fitness of candidate solutions is used to generate new candidates. Their paper has presented results considering one dimensional search space but it can further be extended to higher dimensional space. Liu et al. (2014) have proposed an Opposite Center Learning (OCL), where they have defined opposite point as the optimum solution among a pairwise sampling of search space given a random initial point. This method accelerates the convergence of population-based search algorithms. Rahnamayan, Jesuthasan, Bourennani, Salehinejad, and Naterer (2014) have suggested an Oppositional Target-Domain Estimation (OTDS) where they divided the search space into grids, making the estimation of the target domain faster. However, the computational complexity increases if the solution lies adjacent to a grid border.

Literature survey reveals that there are only few works available where researchers have used OBL alongside GWO. The elite opposition-based learning (EOGWO) (Zhang, Luo & Zhou, 2017) performs a simplex based opposition on all the wolves. Instead of taking the upper and lower limits of the function, opposition is done using the limits of all the wolves. In Improved Grey Wolf Optimizer (IGWO) (Nasrabadi, Sharafi, & Tayari, 2016), the wolves are divided into two sub-groups and the best wolves of one sub-group replaces the worst wolves of the other sub-group. In the two parts, one part is partially opposed (in one dimension) and the rest are opposed in all dimensions. With iterations, the size of the full opposed wolves is decreased.

Some authors have included chaos as well as OBL in their methods. In 2018, Gupta and Deep (2018) have proposed an opposition based chaotic GWO where they have introduced OBL and local chaotic search to the GWO to increase the exploration power of the algorithm. This method shows better convergence as well as better avoidance of local minima than GWO. In 2018, Ibrahim, Elaziz, and Lu (2018) have introduced a chaotic opposition based GWO with DE and disruption operator to solve global optimization problem. Here, the initial population is enhanced with the help of chaotic OBL and later is repeatedly updated using DE operators alongside GWO. In this algorithm, GWO serves for better exploration whereas DE operators help avoiding the local optima providing better exploitation. To balance between these two, disruptive operator is used, so that the population remains diverse after each iteration.

Pradhan, Roy, and Pal (2017) have combined the concept of GWO and OBL, for accelerating the convergence rate of the algorithm, and then used it to resolve the Economic Load Dispatch (ELD) problem of power system optimally. On applying this algorithm to solve any type of ELD problem, it has showed competitive convergence compared to GWO. Another field where OBL and GWO have been applied is parallel machine scheduling in cloud computing (Natesan and Chokkalingam, 2017).

In all these papers, the opposite of a candidate is found for all dimensions or in 1 randomly chosen dimension. However, the approach of only opposing certain dimensions of a solution to enhance the result has not been attempted before. Moreover, to allow for a faster convergence, OBL has been applied taking into mind the extremum present in the current set of wolves. This essentially binds the solutions created using OBL within the current range of the population. The opposition-based GWO technique introduced in our proposed method can be enhanced by hybridizing it with other swarm intelligence algorithms or EAs discussed above for better optimal results and convergence.

3. Basic theory and notations

The two key topics related to our work namely GWO and OBL are discussed in Sections 3.1 and 3.2 respectively. After that Spearman's coefficient is described in Section 3.3.

3.1. Grey Wolf Optimizer (GWO)

GWO is a metaheuristic optimizer based on the hunting technique and social hierarchy (see Fig. 1) of grey wolves (*canis lupus*), which has been proposed by Mirjalili et al. (2014). In a pack of grey wolves, the leader is called alpha (α), its subordinates in decision-making or pack activities are called beta (β) and the lowest ranking wolves are called omega (ω). If any wolf does not fit in any of the three categories then the wolf is called delta (δ).

During a hunt, the grey wolves encircle the prey, which can be mathematically expressed as the following equations:

$$\vec{D} = \left| \vec{C_1}.\vec{X_p}(it) - \vec{X_w}(it) \right| \tag{3.1}$$

$$\vec{X_w}(it+1) = \vec{X_p}(it) - \vec{C_2}.\vec{D} \tag{3.2}$$

where,

it is the current iteration number,
$\vec{X_p}$ is the position vector of the prey,
$\vec{X_w}$ is the position vector of a grey wolf,
$\vec{C_1}, \vec{C_2}$ are the coefficient vectors which are calculated as:

$$\vec{C_2} = 2\vec{d}.\vec{rv}_1 - \vec{d} \tag{3.3}$$

$$\vec{C_1} = 2.\vec{rv}_2 \tag{3.4}$$

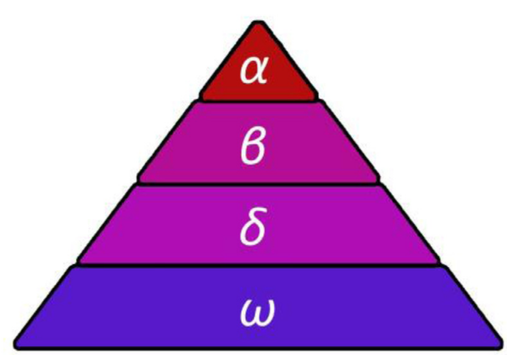


Fig. 1. Social hierarchy of Grey Wolves.

where each element of $\vec{d}$ is linearly decremented from 2 to 0 over iterations to mathematically model the wolves approaching the prey and $\vec{rv}_1, \vec{rv}_2$ are random vectors in the range [0, 1].

In a search space the best candidate solution is α, so it is more likely to be close to the optima and then β and δ respectively. So, to mathematically express the hunting behavior, the best 3 candidate solutions (i.e. $\vec{X}_\alpha, \vec{X}_\beta$ *and* $\vec{X}_\delta$) are taken into consideration and as these three change, we change the other search agents accordingly using the formulas:

$$\vec{D}_\alpha = \left|\vec{C}_{11}.\vec{X}_\alpha(it) - \vec{X}_w(it)\right|$$
$$\vec{D}_\beta = \left|\vec{C}_{12}.\vec{X}_\beta(it) - \vec{X}_w(it)\right| \quad (3.5)$$
$$\vec{D}_\delta = \left|\vec{C}_{13}.\vec{X}_\delta(it) - \vec{X}_w(it)\right|$$

$$\vec{X}_{w1} = \vec{X}_\alpha(it) - \vec{C}_{21}.\vec{D}_\alpha$$
$$\vec{X}_{w2} = \vec{X}_\beta(it) - \vec{C}_{22}.\vec{D}_\beta \quad (3.6)$$
$$\vec{X}_{w3} = \vec{X}_\delta(it) - \vec{C}_{23}.\vec{D}_\delta$$

$$\vec{X}_w(it+1) = \frac{\vec{X}_{w1} + \vec{X}_{w2} + \vec{X}_{w3}}{3} \quad (3.7)$$

The fluctuation range of $\vec{C_2}$ ($\vec{C_2}$ is a random vector in the range $[-2d, 2d]$) is decreased linearly with $\vec{d}$. When the value of $|C_2|$ is less than 1, the wolf in the iteration thereafter will be positioned between optimum and current location. This leads to exploitation. On the other hand, $|C_2| > 1$ suggests divergence from the prey (exploration). The elements of $\vec{C_1}$ are random values in the range of $[0, 2]$. The effect of prey is stochastically emphasized ($C_1 > 1$) or de-emphasized ($C_1 < 1$) through the random weights (C_1) in Eq. (3.1). This randomization in GWO provides necessary exploration as well as helps in avoiding local optima. In (Ozsoydan, 2019b) the authors have demonstrated statistically the importance and effect of the best three candidate solutions on the performance of GWO.

3.2. Opposition Based Learning (OBL)

This scheme was proposed by Tizhoosh (2005), which is fundamentally based on estimates and its counter estimates. Whenever we try to find an optimal solution x of some given objective function, initially we take a random number $\hat{x}$ and then gradually try to minimize the difference between $\hat{x}$ and x.

Generally, an optimization method starts at a random point where initial population or initial parameters used are based on randomness. If the random estimate is close to the optimal solution, the method converges faster. However, the initial guess might be far from the optimum or for the worst case, even they may be at exact opposite position. In that case, it would take considerably greater time to converge or in the worst case, it may not converge at all. As it is impossible to make a best guess without any prior information, we can look for the opposite direction of each candidate solution x in each step and if found beneficial then we can use the opposite point $\tilde{x}$ as the candidate solution before proceeding to the next iteration.

Definition 1. Let x be a real number which is defined on the interval $[a_1, a_2]$. The *opposite number of x i.e.* x_{op} is defined as the equation given below:

$$x_{op} = a_1 + a_2 - x \quad (3.8)$$

If $a_1 = 0$ and $a_2 = 1$, then

$$x_{op} = 1 - x \quad (3.9)$$

Similarly, we can define the opposite number in case there is more than 2 dimensions.

Definition 2. Let $P(x_1, x_2, \ldots, x_n)$ be a point in a n-dimensional coordinate system with $x_1, \ldots, x_n$ are real numbers where each of x_i lies in the range $[a_{1i}, a_{2i}]$. The opposite point of P i.e. $\tilde{P}$ is defined by

$$x_{iop} = a_{1i} + a_{2i} - x_i \quad \forall i \in [1, n] \quad (3.10)$$

Where x_{iop} is coordinate of $\tilde{P}$

Let $f(x)$ be the function in focus and $g(.)$ be the proper evaluation function (e.g. fitness function). Here in Fig. 2, $x \in [a1, b1]$

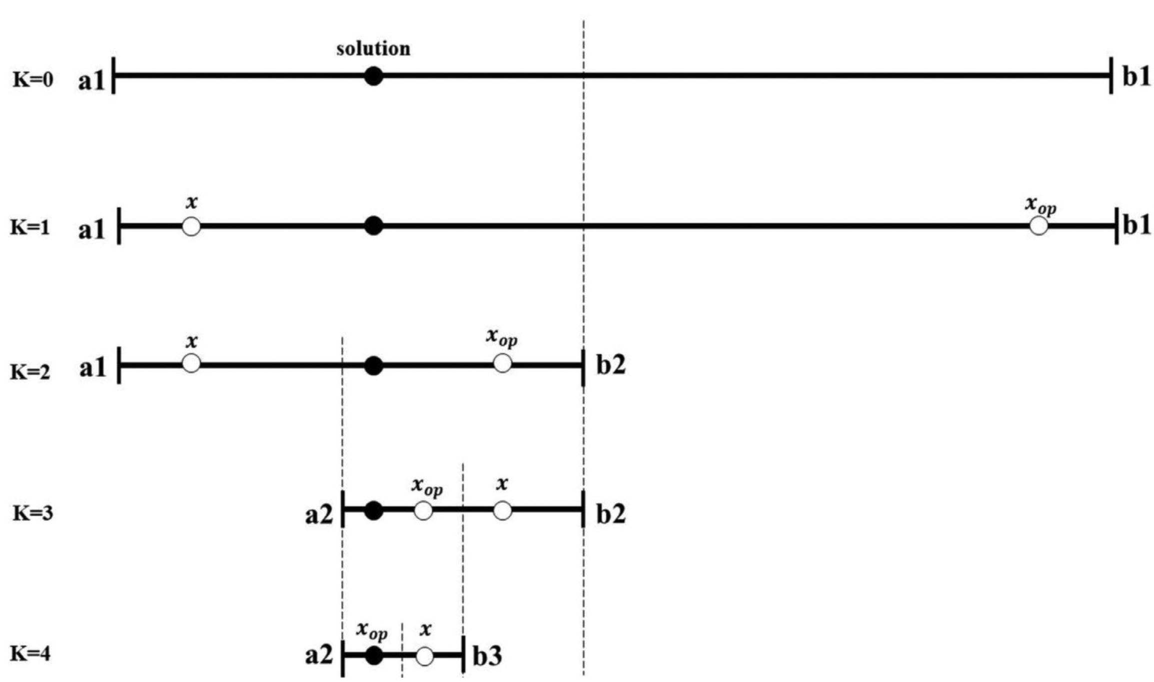


Fig. 2. Solving a one-dimensional equation by dividing the search interval in two halves in each iteration (k=0 to 4) with respect to optimal position of the current estimate $\boldsymbol{x}$ and its opposite $\boldsymbol{x_{op}}$.

be the initial randomized estimate and x_{op} be its opposite, then in each iteration we proceed with x if x is closer to the solution i.e. $g(f(x)) \geq g(f(x_{op}))$, and with x_{op} for any other case. Also, in each iteration, the search space is divided in halves until either one of those is close to the solution.

3.3. Spearman's correlation coefficient

This is a method to find the statistical dependence between two series (here, the positions of two candidate solutions), where each series represents ranking of a number of individuals by an independent judge. This correlation coefficient is identical to the Pearson's Correlation Coefficient between the rank values of two variables.

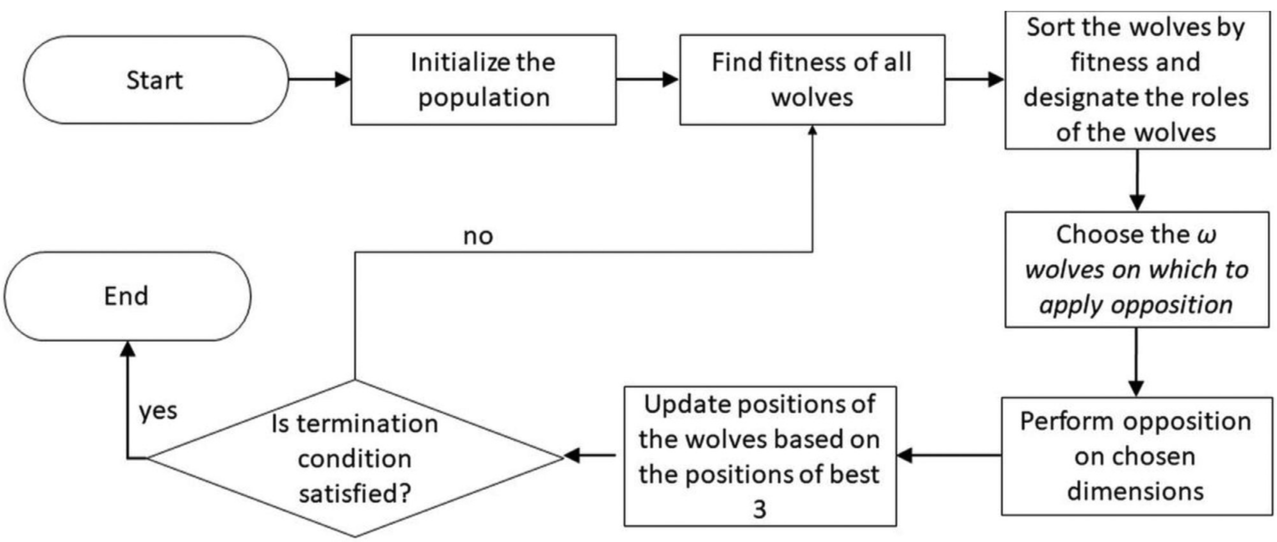
Suppose, we have n individuals and their values in two series are $u_1, u_2, u_3, \ldots, u_n$ and $v_1, v_2, v_3, \ldots v_n$. Let, $d_i = u_i - v_i, i = 1, 2, \ldots n$

Then the Spearman's Correlation Coefficient is given by

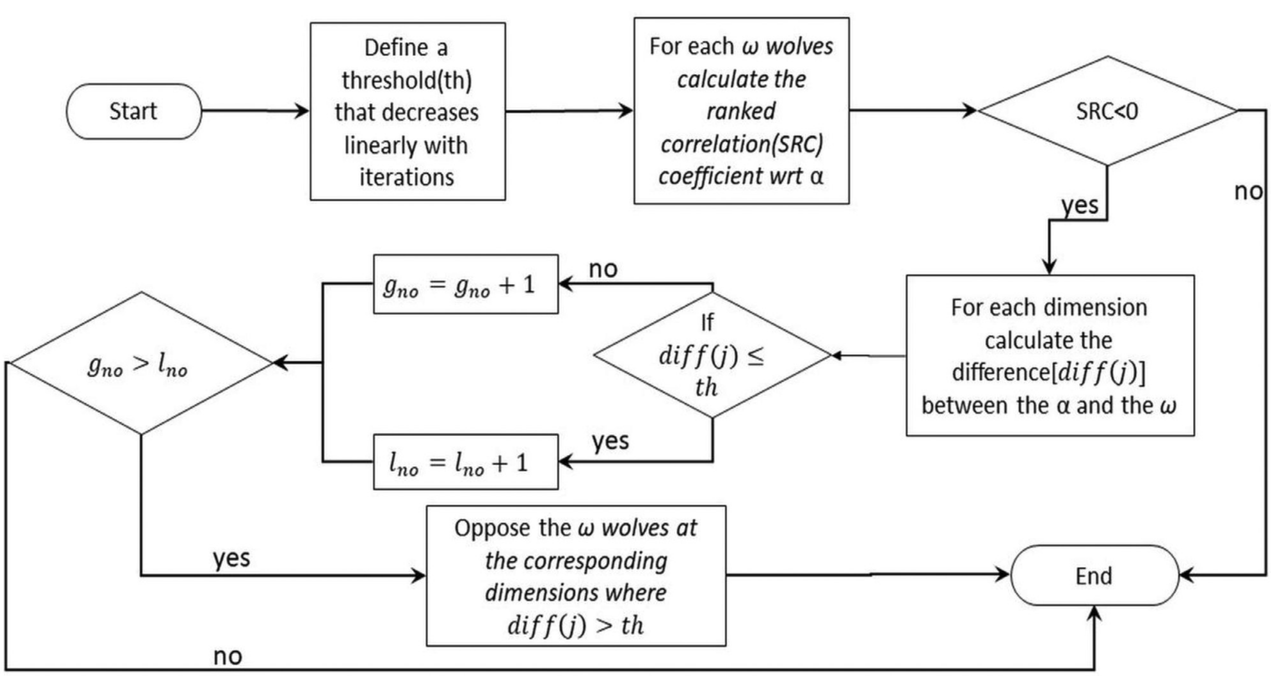
$$r_R = 1 - \frac{6 * \sum_i (d_i)^2}{dim * \left(\dim^2 - 1\right)} \tag{3.11}$$

When there is perfect agreement between two series, association is positively perfect. In that case $u_i = v_i$, for each i and hence $\sum_i d_i^2 = 0$, so $r_R = 1$. Again, in case of disagreement between the two series the correlation coefficient value is less than 1.

In our proposed algorithm, we use this correlation coefficient to measure the associativity between α wolf and each of the ω wolves. Taking α wolf series $u_1, u_2, u_3, \ldots, u_n$ and each of the ω wolves as series $v_1, v_2, v_3, \ldots v_n$ (where n is the dimension of the



(a) Flowchart for the overview of the proposed algorithm



(b) Flowchart for Selection and Opposition of wolves

Fig. 3. Flowchart of the proposed model – SOGWO.

search space) we calculate r_R using Eq. (3.11) to calculate the correlation between them which further leads to the dimensions that need to be opposed. Spearman coefficient is used, as in this case, there is no need to have normal distribution of the values in the wolves.

4. Proposed methodology

In this paper, OBL is used alongside the GWO to improve the efficiency of the later. The flowchart of the proposed algorithm is given in Fig. 3. We have named the proposed algorithm as Selective Opposition based Grey Wolf Optimization (SOGWO). Given the number of search agents, upper bound, lower bound and dimension of the search agents, the population is initialized. The population vectors are selected randomly keeping the given constraints in consideration.

Generating a new candidate solution through opposition allows us to search the opposite position (with respect to the candidate whose opposition is made) of the search space. Keeping this in mind, it can be said that the best solution's (α's in our case) position can be considered as the best one in the search space. Now an ω wolf might be in position which is both near and far away from the α. OBL is applied on the ω wolves which are far away from the α to allow for improvement of less fit wolves. This allows for a more directed use of OBL.

Using selective OBL strategy in GWO, it successfully increases the speed and the probability of convergence to the global optimum (Ibrahim et al., 2018). After every iteration, a new set of candidate solutions is generated, which consists of α, β, δ and ω vectors. Among these solutions, the best candidate is α; then β, δ; and the least fit candidates are ω. Therefore, there may be a possibility that the ω wolves are at opposite position in the search space from the optimum or the vicinity of the optimum. Hence, after every iteration, the ranked correlation coefficient (see Eq. (3.11)) of each of the ω candidate solutions is calculated with respect to the α solution (best solution available to us).

If the correlation coefficient is less than zero i.e. the points have a negative correlation, then the position of that particular ω search agent is opposed. Spearman coefficient helps select the wolves to be opposed and to select the dimensions in which it is to be opposed; thereby limiting and directing our search, and making it more efficient. This is pictorially depicted in Fig. 4.

A threshold variable (*threshold*), initialized as d, is decreased linearly with iteration. This value determines if the dimension of the ω wolf is near the α or not. For each dimension (j) of the given vector (X) and the α vector (X_α), the difference between them is calculated using Eq. (4.1).

$$diff(j) = |X(j) - X_\alpha(j)| \tag{4.1}$$

If the value of *diff*(j) is greater than *threshold*, then the number of dimensions where the difference is greater (g_{no}) is incremented i.e. $g_{no} = g_{no} + 1$. Let, the total dimension be n. If $(n - g_{no}) < g_{no}$, then the dimensions for which the difference between X and X_α are greater than threshold are opposed using the Eq. (4.2).

$$\tilde{X}(i) = ub(i) + lb(i) - X(i) \tag{4.2}$$

where, $i \in \{j: diff(j) \leq threshold\}$, $\tilde{X}$ is the resultant opposed vector, $ub(i)$ is the upper bound of the ith dimension in the current population and similarly $lb(i)$ is the lower bound of the ith dimension in the current population. Generally speaking, OBL is applied using the lower and upper bounds of the function. Let the optimal value of a dimension be $\tilde{x}$, and the current value be $\tilde{x} + \partial$, where ∂ is a very small value. On opposition the value will become $a_i + b_i - \tilde{x} - \partial$ which may not be near the optimal solution unless $\tilde{x} = (a_i + b_i)/2$. So, use of opposition when the algorithm converges can adversely affect it by causing unnecessary exploration. To achieve a balance between exploration and exploitation, the bounds, which are derived from the population, are used for opposition of a candidate.

Then, the selected ω wolves are opposed. If the fitness is better, then the vectors are replaced by the opposition vectors, otherwise, the vectors are kept as they were. The α, β and δ vectors are decided from the modified set of candidate solutions. The pseudo code for the proposed method is given below:

Pseudo code of SOGWO

```
1. Initialize n search agent positions with random numbers X_i(i = 1, 2, ...., n)
2. Initialize α, β and δ positions and α, β and δ fitness scores
3. Initialize iter=0
4. while (iter<maximum iterations)
5.   for i=1 to n
6.     Reinitialize the wolf dimensions that go beyond the boundaries of the search space
7.     Calculate fitness function for each search agent
8.   end for
9.   sort the search agents in increasing order of fitness value (increasing order for minimization problem; decreasing order for maximization problem)
10.   Update α, β and δ positions and their respective fitness scores
11.   a=2 - [iter *(2/maximum iterations)]
12.   threshold=a
13.   for i=1 to number of ω search agents
14.     for j=1 to number of dimensions(dim)
15.       diff(j)=|X(j)-X_α (j)| //difference along jth dimension of α and ω agent
16.       if diff(j)>threshold
17.         g_no=g_no+1
18.       end if
19.     end for
20.     Src= 1 − (6*Σ_j (diff(j))^2)/(dim*(dim^2−1)) //Src = Spearman's rank correlation coefficient
21.     if src<=0
22.       if (dim-g_no<g_no)
23.         for k∈{j:diff(j)>threshold}
24.           X(k)=ub(k)+lb(k)-X(k)
25.         end for
26.       end if
27.     end if
28.   end for
29.   Update the position of each of the search agents including ω's using Eqs. (3.5)--(3.7)
30.   iter = iter + 1
31. end while
32. return position of α
```

Therefore, unbridled exploration is replaced by exploitation as the values converge, thereby allowing for a faster convergence rate. OBL works well in case of non-symmetrical functions as new wolves are generated as reflection of the original wolves. In case of the proposed algorithm this limitation does not hamper performance as we do not oppose all dimensions of the omega (ω) wolf but those with difference more than the threshold with the alpha (α). Moreover, the opposition is decided based on the upper and lower limits of the population not the search space, so this disadvantage is further mitigated.

5. Results and discussion

The proposed algorithm is evaluated on 23 standard optimization functions commonly used in literature (Rashedi et al., 2009). There are three kinds of functions, which are generally used to evaluate any such algorithm consisting of 7 Uni-modal functions, 6 Multi-modal functions and 10 Multi-modal with fixed dimension.

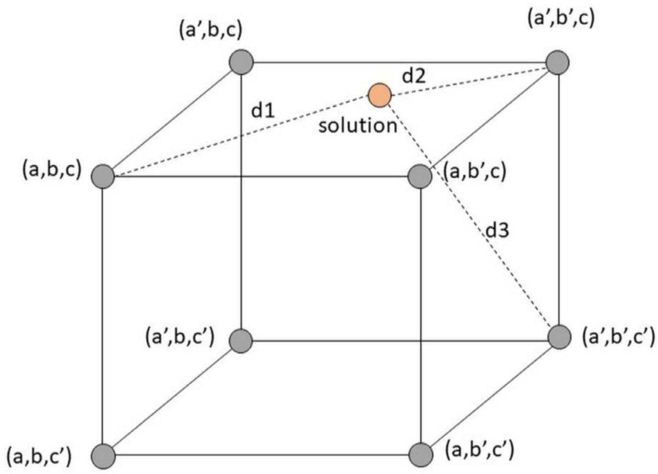


Fig. 4. A hypothetical example showing the usefulness of selective OBL strategy in GWO. Let's suppose in a 3-dimensional space, (a, b, c) be a position vector of a wolf and (a', b', c') be the exact opposite point of (a, b, c). The distance from the solution of the two vectors (a, b, c) and (a', b', c') are d1 and d3 respectively. Also suppose that (a', b', c) be the point where (a, b, c) is opposed along the 1st two dimensions only (selective opposition) and its distance from the solution is d2. From figure, it is clear that d2<d1<d3, i.e. in this case opposition vector diverges from the solution, whereas selective opposition vector gives better result.

Table 1
Parametric details of the algorithms with which proposed algorithm has been compared.

Algorithm	Parameter and its corresponding value
GSA	$c_1 = 2$, $c_2 = 2$, $G_0 = 1$
PSO	$c_1 = 2$, $c_1 = 2$,
GPS	$c_1 = 2$, $c_2 = 2$, $c_3 = 0.5$, $c_4 = 1.5$
PSOGSA	$c_1 = 0.5$, $c_2 = 1.5$
GWO	$\vec{d} = 2$ (linearly decreased over iterations)
EOGWO	$\vec{d} = 2$ (linearly decreased over iterations)
IGWO	$\vec{d} = 2$ (linearly decreased over iterations)
COA	$nfeval_{max} = 20000$, $n_p = 20$, $n_c = 5$

The results are compared with the other state-of-the-art methods along with its parent algorithm i.e. GWO. The comparative algorithms are GSA, PSO, GPS, PSOGSA, GWO, EOGWO, IGWO and COA. The population size of each algorithm is taken as 50. The number of iterations in each algorithm is 1000. Each algorithm is run for 25 times and the best, average (avg) and standard deviation (std) of the results are taken. The best results are considered to be the ones with the lowest value of average. If the value of average is same, the algorithm with lower value of standard deviation is considered as better. The parametric details of the compared algorithms are in Table 1.

The results are provided in Tables 2–4, where best results are marked in bold. Out of 23 functions, SOGWO performs better in 10 cases. As can be seen from the results, in 16 cases SOGWO outperforms GWO, which shows the effectiveness of using OBL to enhance exploration in GWO. The results show that across all categories, SOGWO outperforms GWO giving credibility to our assumption that our model allows for a faster convergence and more exploration. SOGWO also gives better results than most state-of-the-art techniques. Comparison of SOGWO with other OBL techniques applied on grey wolf - EOGWO and IGWO, also shows that the proposed SOGWO performs better. From these results, it can be stated that targeted opposition of specific wolfs yields a far better result than opposition of all wolves or half of the wolves. SOGWO even outperforms COA in all cases. The convergence of the algorithms and the shape of the corresponding functions (in 2D) are depicted in Figs. 5–7.

5.1. Significance test

Friedman test is performed here for determining the statistical significance of SOGWO results over the other 7 algorithms (which were used for comparison in Tables 2 and 4). Friedman test is a non-parametric statistical test which provides the measure of the differences between multiple methods. We have considered the null hypothesis that there is no significance difference in the results of the algorithms at 0.1% significance level. The probability distribution of the test statistics (**H**) that is generated by Friedman's test can be approximated as a Chi-squared distribution. From the test statistics, we can find the p-value suggesting the probability of the cases where null hypothesis cannot be rejected. From the test results, we find that the p-value is <0.1. Therefore, from the results, we can clearly say that the null hypothesis is rejected. This as well clearly ratifies the presence of at least one set of significant results.

To confirm whether our algorithm provides significant result or not, we also perform Wilcoxon test as a post-hoc multiple com-

Table 2
Results of SOGWO and other state-of-the art methods on 7 Uni-modal functions.

Function	Uni-modal Functions									
	Value Heads	GSA	PSO	GPS	PSOGSA	GWO	EOGWO	IGWO	COA	SOGWO
F1	best	1.1000E−17	1.1000E−15	6.6000E−19	3.2900E−19	9.0700E−73	1.7600E−73	3.3560E−76	1.0300E -12	3.8110E−79
	avg	2.0000E−17	1.3000E−11	1.2000E−18	4.7400E−19	1.3600E−70	2.8100E−71	4.7535E−73	1.2040E−10	**6.0467E−77**
	std	5.5000E−18	8.8000E−11	3.0000E−19	8.0400E−19	2.5700E−70	8.4600E−71	9.5335E−73	1.0270E−10	**1.4851E−76**
F2	best	1.4000E−08	4.4000E−09	3.3000E−09	2.4700E−09	3.5900E−42	3.0000E−42	3.7722E−43	1.0308-10	3.5047E−46
	avg	2.4000E−08	2.9000E−06	5.2000E−09	2.9300E−09	5.6400E−41	2.6300E−41	4.3196E−42	1.0056E−09	**1.1778E−44**
	std	4.4000E−09	1.3000E−05	9.0000E−10	2.6400E−10	6.4300E−41	2.7300E−41	7.8752E−42	4.5361E−08	**1.3421E−44**
F3	best	7.5000E+01	1.9000E+01	3.1000E+00	2.9200E+02	2.5600E−25	1.0100E−26	5.0001E−26	3.3835E+09	1.1677E−28
	avg	2.3000E+02	1.2000E+02	9.7000E+01	1.8200E+03	1.0900E−19	1.2900E−20	1.5218E−20	3.3835E+09	**5.3966E−22**
	std	1.0000E+02	7.5000E+01	1.1000E+02	4.8200E+02	3.1100E−19	3.4000E−20	4.0202E−20	0.0000E+00	**2.5958E−21**
F4	best	2.1000E−09	1.4000E−01	8.2000E−10	1.3000E+01	1.2800E−18	1.2600E−19	4.8137E−20	1.0000E+02	7.0819E−21
	avg	6.4000E−02	4.2000E−01	1.3000E+00	2.2000E+01	1.9400E−17	2.3900E−17	8.0697E−19	1.0000E+02	**1.1803E−19**
	std	2.5000E−01	1.9000E−01	9.8000E−01	3.8900E+00	3.6800E−17	5.3300E−17	1.1133E−18	0.0000E+00	**1.5101E−19**
F5	best	2.6000E+01	2.5000E+01	2.3000E+01	1.6000E+01	2.5100E+01	2.5900E+01	2.5052E+01	7.4934E+09	2.4994E+01
	avg	2.8000E+01	2.7000E+01	2.6000E+01	**2.6000E+01**	2.6300E+01	2.6700E+01	2.6341E+01	7.4934E+09	2.6489E+01
	std	1.0000E+01	8.4000E+00	8.8000E+00	**2.5000E+00**	6.6900E−01	6.1400E−01	7.3644E−01	0.0000E+00	7.6235E−01
F6	best	7.4000E−18	8.3000E−16	6.0000E−19	3.3000E−19	1.0900E−05	8.0900E−06	1.2677E−05	1.0100E+06	6.1853E−06
	avg	1.9000E−17	1.3000E−12	1.2000E−18	**5.0500E−19**	4.1200E−01	2.1500E−01	3.2903E−01	1.0100E+06	2.8281E−01
	std	6.4000E−18	7.1000E−12	3.3000E−19	**9.4000E−20**	2.4500E−01	2.3000E−01	2.4595E−01	0.0000E+00	2.4746E−01
F7	best	8.4000E−03	1.7000E−03	1.1000E−03	1.5000E−02	1.4900E−04	1.3100E−04	1.4084E−04	1.3556E+04	8.0573E−05
	avg	2.8000E−02	7.0000E−03	3.1000E−03	3.3000E−02	5.6800E−04	**4.3400E−04**	6.0719E−04	1.3556E+04	4.9306E−04
	std	1.7000E−02	2.5000E−03	1.2000E−03	9.3000E−03	3.5400E−04	**2.7700E−04**	4.3279E−04	8.0943E−05	2.7143E−04

Table 3
Results of SOGWO and other state-of-the art methods on 6 Multi-modal functions.

Function	Multi-modal functions									
	Value Heads	GSA	PSO	GPS	PSOGSA	GWO	EOGWO	IGWO	COA	SOGWO
F8	best	−4.2000E+03	−1.0000E+04	−8.9000E+03	−8.8500E+03	−7.0900E+03	−7.7900E+03	−7.6008E+03	1.8059E+04	−8.1750E+03
	avg	−2.7000E+03	**−9.0000E+03**	−7.5000E+03	−8.0900E+03	−6.0700E+03	−6.2700E+03	−6.3096E+03	1.8059E+04	−6.5680E+03
	std	4.7000E+02	**5.2000E+02**	7.7000E+02	4.7100E+02	5.3700E+02	7.7100E+02	9.8077E+02	3.8348E−12	8.0271E+02
F9	best	9.0000E+00	1.8000E+01	9.0000E+00	4.4800E+01	0.0000E+00	0.0000E+00	0.0000E+00	2.8925E+03	0.0000E+00
	avg	1.7000E+01	4.1000E+01	2.1000E+01	7.4200E+01	5.2000E+00	5.2900E−01	**0.0000E+00**	2.8925E+03	**0.0000E+00**
	std	4.3000E+00	1.5000E+01	6.1000E+00	1.1000E+01	1.8900E+00	1.9400E+00	**0.0000E+00**	4.7935E−13	**0.0000E+00**
F10	best	2.2000E−09	4.6000E−09	5.4000E−10	4.3200E−10	7.9900E−15	7.9900E−15	7.9936E−15	1.9967E+01	8.8818E−16
	avg	3.4000E−09	9.1000E−08	8.8000E−10	5.0700E−10	1.3100E−14	1.4000E−14	1.1120E−14	1.9967E+01	**8.8818**E−**16**
	std	4.1000E−10	2.0000E−07	1.3000E−10	4.7000E−11	2.7300E−15	3.2000E−15	3.7430E−15	0.0000E+00	**0.0000E+00**
F11	best	2.0000E+00	5.1000E−15	0.0000E+00	2.8600E−06	0.0000E+00	0.0000E+00	0.0000E+00	9.0010E+03	0.0000E+00
	avg	4.3000E+00	1.2000E−02	2.3000E−02	2.3300E−01	5.2300E−04	1.6800E−03	**0.0000E+00**	9.0010E+03	**0.0000E+00**
	std	1.6000E+00	1.2000E−02	3.0000E−02	3.5000E−01	2.6100E−03	4.8000E−03	**0.0000E+00**	0.0000E+00	**0.0000E+00**
F12	best	6.2000E−20	1.6000E−18	4.7000E−21	1.0100E+00	6.5700E−03	1.7200E−06	1.3150E−02	2.5600E+10	2.6183E−02
	avg	2.5000E−02	**1.5000**E−**02**	5.0000E−02	4.4600E+00	2.6600E−02	2.2900E−02	3.2924E−02	2.5600E+10	5.6082E−02
	std	6.1000E−02	**3.6000**E−**02**	1.3000E−01	1.8000E+00	1.5500E−02	1.8500E−02	1.7568E−02	0.0000E+00	1.4242E−02
F13	best	1.2200E−18	9.9000E−131	3.7500E−08	9.2900E−20	1.5800E−05	1.5800E−05	8.9404E−02	4.1006E+10	1.4184E−05
	avg	2.1000E−18	**2.0000**E−**31**	8.4800E−02	2.2000E−03	3.2500E−01	2.5700E−01	3.4626E−01	4.1006E+10	3.5289E−01
	std	5.0000E−19	**4.3000**E−**31**	8.0000E−02	4.5000E−03	1.5600E−01	1.6400E−01	1.7283E−01	0.0000E+00	1.2815E−01

Table 4
Results of SOGWO and other state-of-the art methods on 10 Multi-modal functions with fixed dimensions.

Function	Multi-modal functions with fixed dimensions									
	Value Heads	GSA	PSO	GPS	PSOGSA	GWO	EOGWO	IGWO	COA	SOGWO
F14	best	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	9.9800E−01	9.9800E−01	9.9800E−01	5.0000E+02	9.9800E−01
	avg	3.8000E+00	**1.0000E+00**	1.0000E+00	1.3900E+00	3.1100E+00	3.8200E+00	2.8779E+00	5.0000E+02	3.4280E+00
	std	2.6000E+00	**3.2000**E−**17**	5.8000E−01	8.7500E−01	3.7300E+00	3.8600E+00	3.5872E+00	5.9918E−14	3.7209E+00
F15	best	1.4000E−03	3.1000E−04	3.1000E−04	3.1000E−04	3.0700E−04	3.0700E−04	3.0749E−04	4.1020E+01	3.0749E−04
	avg	4.1000E−03	1.2000E−03	4.1000E−04	4.1000E−04	4.3600E−03	5.2400E−03	**3.0751**E−**04**	4.1020E+01	2.3811E−03
	std	3.2000E−03	4.0000E−03	3.4000E−04	3.4000E−04	8.1700E−03	8.6800E−03	**4.4401**E−**08**	7.4898E−15	6.0288E−03
F16	best	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0300E+00	−1.0300E+00	−1.0316E+00	6.4208E+03	−1.0316E+00
	avg	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0000E+00	−1.0200E+00	−1.0200E+00	−1.0200E+00	6.4208E+03	**−1.0316E+00**
	std	4.0000E−16	2.3000E−16	2.8000E−16	2.8000E−16	4.8000E−09	3.4500E−09	4.1291E−09	9.5869E−13	**3.7505**E−**09**
F17	best	4.0000E−01	4.0000E−01	4.0000E−01	3.9800E−01	3.9800E−01	3.9800E−01	3.9789E−01	9.0010E+03	3.9789E−01
	avg	4.0000E−01	4.0000E−01	4.0000E−01	3.9800E−01	3.9800E−01	3.9800E−01	3.9800E−01	9.0010E+03	**3.9700**E−**01**
	std	3.4000E−16	3.4000E−16	3.4000E−16	0.0000E+00	3.3600E−07	4.8200E−07	1.1799E−04	0.0000E+00	**4.8577**E−**07**
F18	best	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	7.6728E+04	3.0000E+00
	avg	3.0000E+00	3.0000E+00	**3.0000E+00**	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	7.6728E+04	3.0000E+00
	std	2.2000E−15	3.1000E−15	**1.6000**E−**15**	8.4000E−16	4.8800E−06	3.6000E−06	1.6890E−06	0.0000E+00	4.6343E−06
F19	best	−3.9000E+00	−3.9000E+00	−3.9000E+00	−3.8600E+00	−3.8600E+00	−3.8600E+00	−3.8628E+00	−3.8628E+00	−3.8628E+00
	avg	−3.6000E+00	**−3.9000E+00**	**−3.9000E+00**	−3.8600E+00	−3.8600E+00	−3.8600E+00	−3.8628E+00	−3.8628E+00	−3.8615E+00
	std	3.0000E−01	**3.1000**E−**15**	**3.1000**E−**15**	2.1900E−15	1.0500E−03	2.3600E−03	3.5874E−06	4.5473E−13	2.7141E−03
F20	best	−3.3000E+00	−3.3000E+00	−3.3100E+00	−3.3200E+00	−3.3200E+00	−3.3200E+00	−3.3220E+00	−3.3220E+00	−3.3220E+00
	avg	−1.9000E+00	−3.3000E+00	−3.3000E+00	**−3.3100E+00**	−3.2500E+00	−3.2700E+00	−3.2615E+00	−3.3219E+00	−3.2663E+00
	std	5.4000E−01	5.5000E−02	2.4000E−02	**6.0700**E−**01**	7.0400E−02	7.5500E−02	6.5470E−02	6.6586E−05	7.3711E−02
F21	best	−5.1000E+00	−1.0000E+01	−1.0000E+01	−1.0200E+01	−1.0200E+01	−1.0200E+01	−1.0153E+01	−1.0153E+01	−1.0153E+01
	avg	−5.1000E+00	−7.2000E+00	−8.5000E+00	−6.1700E+00	**−9.9500E+00**	−9.1400E+00	−8.9366E+00	−1.0153E+01	−9.6556E+00
	std	7.4000E−03	3.3000E+00	3.1000E+00	3.7400E+00	**1.0100E+00**	2.0700E+00	2.2089E+00	1.6314E−04	1.5084E+00
F22	best	−1.0000E+01	−1.0000E+01	−1.0000E+01	−1.0400E+01	−1.0400E+01	−1.0400E+01	−1.0403E+01	−1.0403E+01	−1.0403E+01
	avg	−7.5000E+00	−9.1000E+00	−1.0000E+01	−8.8700E+00	−1.0200E+01	−1.0200E+01	−1.0192E+01	−1.0403E+01	**−1.0403E+01**
	std	2.7000E+00	2.8000E+00	7.2000E−15	3.1400E+00	1.0500E+00	1.0500E+00	1.0549E+00	4.4280E−04	**2.6566**E−**04**
F23	best	−1.1000E+01	−1.1000E+01	−1.1000E+01	−1.0500E+01	−1.0500E+01	−1.0500E+01	−1.0536E+01	−1.0536E+01	−1.0536E+01
	avg	−1.0000E+01	−9.4000E+00	−1.0000E+01	−7.9000E+00	−1.0300E+01	−1.0200E+01	**−1.0536E+01**	−1.0536E+01	−1.0482E+01
	std	7.8000E−01	2.8000E+00	1.6000E+00	3.6900E+00	1.0800E+00	1.6200E+00	**2.1975**E−**04**	9.4523E−05	5.4076E−01

The proposed algorithm is run on dimensions 50 and 100 for the Uni-modal (F1-F7) and Multi-modal (F8-F13) functions and the results have been tabulated in Tables 5 and 6 respectively.

parison test. It is a non-parametric statistical significance test used for pairwise comparison. The null hypothesis for this test is, two sets of results have same distribution. If there is any significant difference in the distributions of the two test results, the p-value generated from the test statistics will be <0.1 (as we perform this test at 0.1% significance level) resulting in the rejection of the null hypothesis i.e. the distributions of the two results are significantly different. From the test results, we can understand that our algorithm provides statistically significant result in some of the cases. The results of these two tests are tabulated in Table 7.

Kruskal–Walli's test is performed to prove the consistency of our method. We run the proposed method 20 times and get the corresponding results. This null hypothesis (the treatments come from the same distribution) is tested at 0.1% significance level. A significant Kruskal Walli's test suggests that the null hypothesis is rejected i.e. at least one of the results is significantly different from another result. The distribution of the test statistics generated can be approximated as the Chi-squared distribution with g-1 degrees of freedom (where g is the number of treatments) and p-value is generated. From the results, it is found that the p-value is >0.1 i.e. the Kruskal-Walli's test fails to reject the null hypothesis. Hence, we can say that the results come from same normal distribution as well as their mean rank is same.

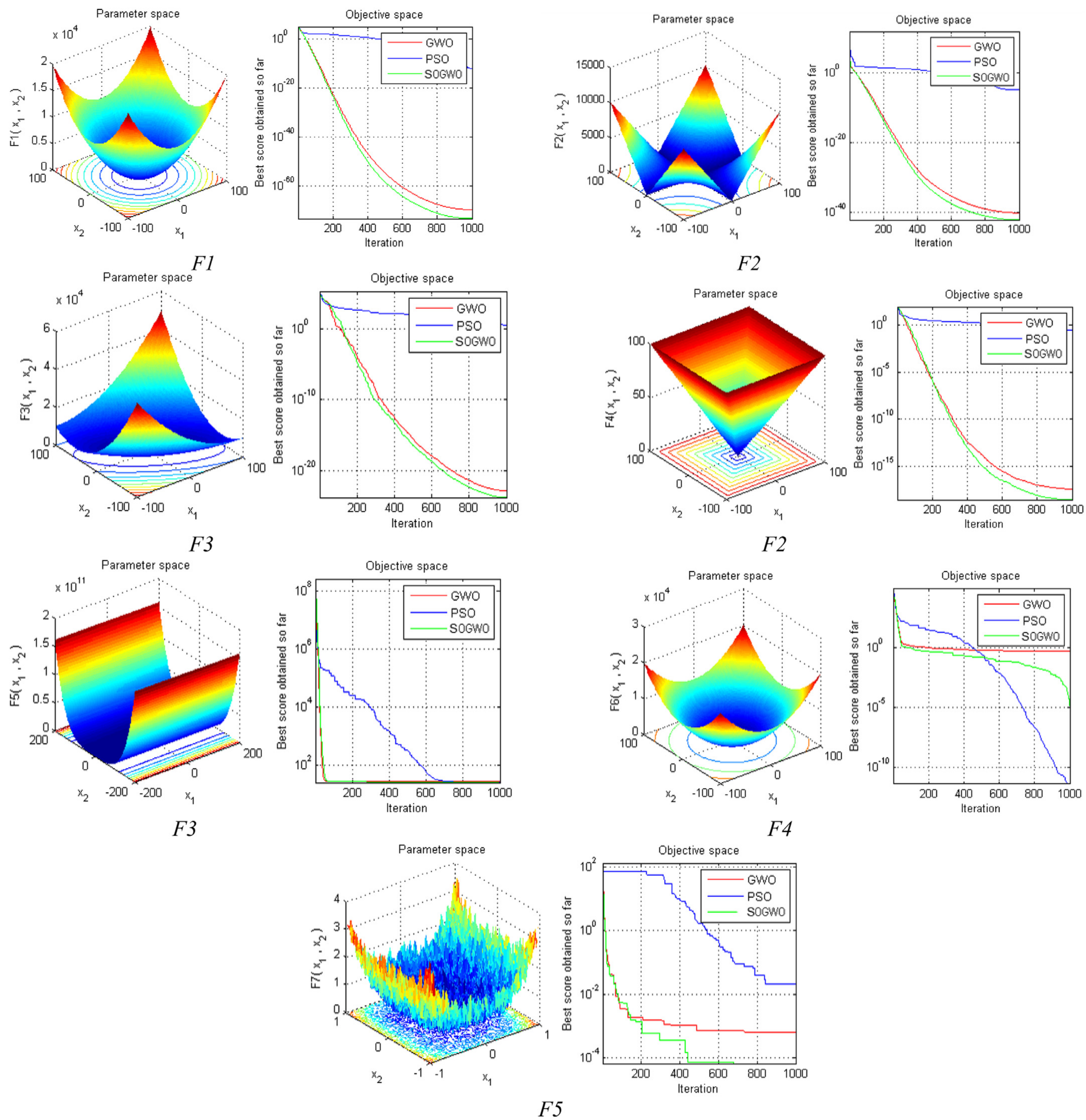


Fig. 5. 2-D version of Uni-modal functions (functions: 1–7) and comparative study of PSO, GWO and SOGWO in the objective place for each function.

Additionally, we have performed a post-hoc multiple comparison test (Dunn's test with Bonferroni's correction) to determine if any of the treatments is statistically significant than other or not. The null hypothesis is tested with 0.1% significance level. A p-value less than 0.1 suggests that the test rejects the null hypothesis (shown as '0' in Table 8) and a p-value greater than 0.1 suggests that the test fails to reject the null hypothesis (shown as '1' in Table 8). From the results it is clear that not a single pair of treatments gives a significant p-value. Therefore, we can say that the proposed algorithm gives consistent results. This test results for Kruskal–Walli's test and Bonferroni-Dunn's test are tabulated in Table 8.

5.2. Result analysis

The Uni-modal functions do not have any local minima. Therefore, these are well suited to check the exploitation ability of the algorithm. From the results of Table 2, it is found that the proposed algorithm gives best results in 4 out of 7 benchmark functions and also gives very competitive results for the rests. Moreover, it outperforms GWO in 6 out of 7 cases.

The Multi-modal functions have multiple number of local optima. The number of the local optima varies exponentially with the number of dimensions. So, these are well suited to check whether the exploration ability of the algorithm is good or not. Form the

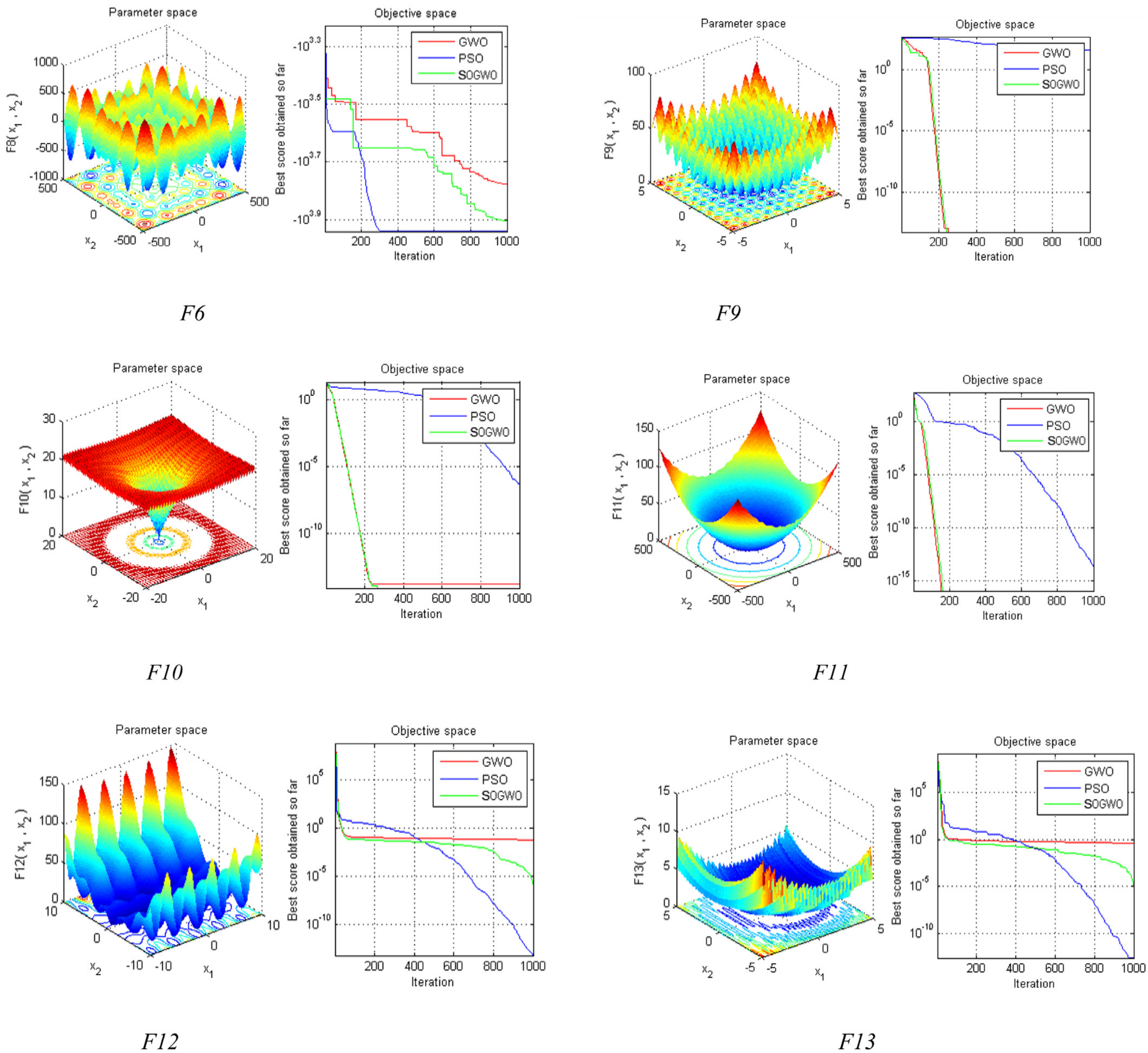


Fig. 6. 2-D version of Multi-modal functions (functions: 8–13) and comparative study of PSO, GWO and SOGWO in the objective place for each function.

Table 5
Comparison of results of SOGWO for dimensions 30, 50 and 100 on Uni-modal functions.

Unimodal Functions	Value Heads	Dim 30	Dim 50	Dim 100
F1	best	3.81E−79	2.49E−53	2.40E−35
	avg	6.05E−77	1.35E−52	1.00E−34
	std	1.49E−76	1.87E−52	7.63E−35
F2	best	3.50E−46	9.59E−32	2.99E−21
	avg	1.18E−44	8.68E−31	6.94E−21
	std	1.34E−44	8.55E−31	3.87E−21
F3	best	1.17E−28	2.13E−13	1.79E−02
	avg	5.40E−22	4.21E−09	1.73E+00
	std	2.60E−21	1.32E−08	1.74E+00
F4	best	7.08E−21	2.26E−12	2.66E−06
	avg	1.18E−19	2.92E−11	6.72E−05
	std	1.51E−19	3.74E−11	1.26E−04
F5	best	2.50E+01	4.60E+01	9.61E+01
	avg	2.65E+01	4.67E+01	9.72E+01
	std	7.62E−01	8.47E−01	8.52E−01
F6	best	6.19E−06	5.01E−01	6.33E+00
	avg	2.83E−01	1.45E+00	7.82E+00
	std	2.47E−01	4.93E−01	7.87E−01
F7	best	8.06E−05	4.12E−04	7.81E−04
	avg	4.93E−04	9.40E−04	2.14E−03
	std	2.71E−04	3.44E−04	6.98E−04

Table 6
Comparison of results of SOGWO for dimensions 30, 50 and 100 on Multi-modal functions.

Multimodal Functions	Value Heads	Dim 30	Dim 50	Dim 100
F8	best	−8.18E+03	−1.12E+04	−2.11E+04
	avg	−6.57E+03	−9.84E+03	−1.48E+04
	std	8.03E+02	7.63E+02	5.57E+03
F9	best	0.00E+00	0.00E+00	0.00E+00
	avg	0.00E+00	7.35E−01	6.47E−01
	std	0.00E+00	2.33E+00	2.05E+00
F10	best	8.88E−16	1.51E−14	5.77E−14
	avg	8.88E−16	2.54E−14	6.87E−14
	std	0.00E+00	5.41E−15	7.00E−15
F11	best	0.00E+00	0.00E+00	0.00E+00
	avg	0.00E+00	0.00E+00	0.00E+00
	std	0.00E+00	0.00E+00	0.00E+00
F12	best	2.62E−02	3.54E−02	1.20E−01
	avg	5.61E−02	6.88E−02	1.73E−01
	std	1.42E−02	2.99E−02	3.30E−02
F13	best	1.42E−05	1.17E+00	5.13E+00
	avg	3.53E−01	1.45E+00	5.44E+00
	std	1.28E−01	1.65E−01	2.55E−01

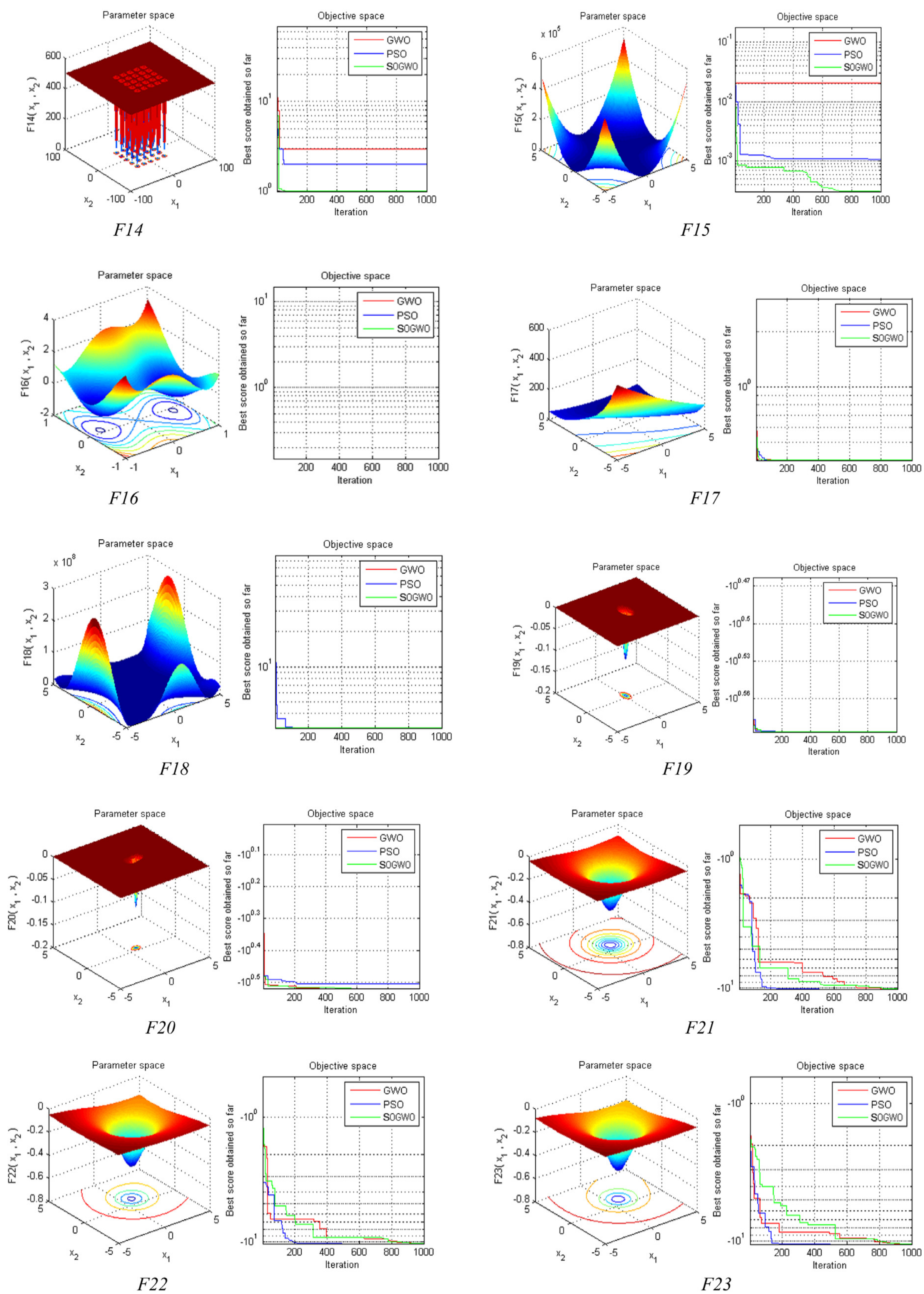


Fig. 7. 2-D version of Multi-modal functions with fixed dimension (functions: 14–23) and comparative study of PSO, GWO and SOGWO in the objective place for each function.

Table 7
Comparison of the results of the proposed method with all the other methods considered here using Friedman test and Wilcoxon test.

Friedman Test Result: H = 42.3981762917, p-value = 1.1400447253e-06								
	Wilcoxon Test Result							
	GSA	PSO	GPS	PSOGSA	GWO	EOGWO	COA	IGWO
SOGWO	**H** = 30.0, **pvalue** = 0.0017306761	**H** = 99.0, **pvalue** = 0.3719614039	**H** = 114.0, **pvalue** = 0.6848744008	**H** = 87.0, **pvalue** = 0.19970475388	**H** = 83.0, **pvalue** = 0.15787466009	**H** = 59.0, **pvalue** = 0.028420589569	**H** =18.0, **pvalue** =0.0004274598	**H** = 90.0, **pvalue** = 0.5754862281

Table 8
Result comparison among 20 runs of the proposed method using Kruskal-Walli's test and Bonferroni-Dunn's test. 1 in the table entry suggests that the test fails to reject the null hypothesis and −1 suggests that no comparison is possible (for same set of values).

Kruskal-Walli's Test Result: H = 0.1800869243, p-value = 0. 0.9999824378																				
Bonferroni-Dunn's Test Result																				
Treatments	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	−1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	−1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	−1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	−1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	−1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	−1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	−1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	−1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	−1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	−1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	−1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	−1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	−1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	−1

results in Tables 3 and 4, it can be seen that the proposed algorithm provides best results in 6 cases and gives very competitive results for the rests. It also outperforms GWO in 12 out of 18 cases.

In Figs. 5–7, the convergence curves for PSO, GWO and SOGWO are placed in same objective space to compare their convergence behavior. From the figures, it is quite assuring that the proposed algorithm depicts a satisfactory convergence behavior and outstrips the convergence rate of other two algorithms in most of the cases.

6. Conclusion

Metaheuristics have been widely used in both science and industry. In recent times, a lot of new algorithms have been proposed mimicking natural phenomenon or natural behavior of animals like foraging. GWO is one of the most emerging algorithms proposed in this space lately. The use of OBL to enhance results of metaheuristics has been attempted before. In this work, however, we adopted a more directed approach to efficiently use OBL in GWO.

Instead of applying opposition to wolves indiscriminately, the ω wolves in the population are evaluated and the most probable ones to undergo improvement are opposed. The use of such targeted opposition helps GWO to improve the exploration rate without maintaining its fast convergence rate. The results show that such an approach allows GWO to perform better than many state-of-the-art algorithms. The use of SOGWO on other real-world applications of optimization can be explored in future. Also, variants of SOGWO to solve multi-objective problems are also suggested. Applying OBL to a portion of the population can be used in other metaheuristics as well.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Credit authorship contribution statement

Souvik Dhargupta: Conceptualization, Methodology, Writing - original draft, Software, Formal analysis. **Manosij Ghosh:** Methodology, Software, Writing - original draft, Formal analysis. **Seyedali Mirjalili:** Writing - review & editing, Formal analysis. **Ram Sarkar:** Supervision, Conceptualization, Writing - review & editing.

References

Abedinpourshotorban, H., Shamsuddin, S. M., Beheshti, Z., & Jawawi, D. N. A. (2016). Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation, 26*, 8–22.

Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications, 36*, 6843–6853 (3 PART 2). doi:10.1016/j.eswa.2008.08.022.

Baykasoğlu, A., & Ozsoydan, F. B. (2018). Dynamic optimization in binary search spaces via weighted superposition attraction algorithm. *Expert Systems with Applications, 96*, 157–174.

Baykasoğlu, A., Ozsoydan, F. B., & Senol, M. E. (2018). Weighted superposition attraction algorithm for binary optimization problems. *Operational Research*, 1–27.

de Vasconcelos Segundo, E. H., Mariani, V. C., & dos Santos Coelho, L. (2019). Design of heat exchangers using Falcon Optimization Algorithm. *Applied Thermal Engineering, 156*, 119–144.

Ebrahimi, A., & Khamehchi, E. (2016). Sperm whale algorithm: An effective metaheuristic algorithm for production optimization problems. *Journal of Natural Gas Science and Engineering, 29*, 211–222.

Ergezer, M., & Simon, D. (2014). Mathematical and experimental analyses of oppositional algorithms. *IEEE Transactions on Cybernetics, 44*(11), 2178–2189.

Ghosh, M., Begum, S., Sarkar, R., Chakraborty, D., & Maulik, U. (2019). Recursive Memetic Algorithm for gene selection in microarray data. *Expert Systems with Applications, 116*, 172–185. doi:10.1016/j.eswa.2018.06.057.

Ghosh, M., Kundu, T., Ghosh, D., & Sarkar, R. (2019). Feature selection for facial emotion recognition using late hill-climbing based memetic algorithm. *Multimedia Tools and Applications*. doi:10.1007/s11042-019-07811-x.

Ghosh, M., Malakar, S., Bhowmik, S., Sarkar, R., & Nasipuri, M. (2019). Feature selection for handwritten word recognition using memetic algorithm. In *In* advances in intelligent computing (pp. 103–124). Springer.

Gupta, S., & Deep, K. (2018). An opposition-based chaotic Grey Wolf Optimizer for global optimisation tasks. *Journal of Experimental & Theoretical Artificial Intelligence*, 1–29.

Ibrahim, R. A., Elaziz, M. A., & Lu, S. (2018). Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Systems with Applications, 108*, 1–27.

Jaddi, N. S., Alvankarian, J., & Abdullah, S. (2017). Kidney-inspired algorithm for optimization problems. *Communications in Nonlinear Science and Numerical Simulation, 42*, 358–369.

Jahani, E., & Chizari, M. (2018). Tackling global optimization problems with a novel algorithm–Mouth Brooding Fish algorithm. *Applied Soft Computing, 62*, 987–1002.

Kaveh, A., & Bakhshpoori, T. (2016). Water evaporation optimization: A novel physically inspired optimization algorithm. *Computers & Structures, 167*, 69–85.

Kaveh, A., & Khayatazad, M. (2012). A new meta-heuristic method: Ray optimization. *Computers & Structures, 112*, 283–294.

Kennedy, J. (2011). *Particle swarm optimization. In* Encyclopedia of machine learning (pp. 760–766). Springer.

Klein, C. E., & dos Santos Coelho, L. (2018). Meerkats-inspired Algorithm for Global Optimization Problems. *ESANN*.

Klein, C. E., Mariani, V. S., & dos Santos Coelho, L. (2018). Cheetah based optimization algorithm: A novel swarm intelligence paradigm. *ESANN*.

Li, M. D., Zhao, H., Weng, X. W., & Han, T. (2016). A novel nature-inspired algorithm for optimization: Virus colony search. *Advances in Engineering Software, 92*, 65–88.

Liu, H., Wu, Z., Li, H., Wang, H., Rahnamayan, S., & Deng, C. (2014). Rotation-based learning: A novel extension of opposition-based learning. In *Pacific rim international conference on artificial intelligence* (pp. 511–522). Springer.

Liu, J., & Lampinen, J. (2005). A fuzzy adaptive differential evolution algorithm. *Soft Computing, 9*(6), 448–462. doi:10.1007/s00500-004-0363-x.

Mahdavi, S., Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. *Swarm and Evolutionary Computation, 39*, 1–23.

Malakar, S., Ghosh, M., Bhowmik, S., Sarkar, R., & Nasipuri, M. (2019). A GA based hierarchical feature selection approach for handwritten word recognition. *Neural Computing and Applications*, 1–20. doi:10.1007/s00521-018-3937-8.

Mirjalili, S., & Hashim, S. Z. M. (2010). A new hybrid PSOGSA algorithm for function optimization. In *Proceedings of ICCIA 2010 - 2010 international conference on computer and information application: 1* (pp. 374–377). doi:10.1109/ICCIA.2010.6141614.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

Mortazavi, A., Toğan, V., & Nuhoğlu, A. (2018). Interactive search algorithm: A new hybrid metaheuristic optimization algorithm. *Engineering Applications of Artificial Intelligence, 71*, 275–292.

Muthiah-Nakarajan, V., & Noel, M. M. (2016). Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. *Applied Soft Computing, 38*, 771–787.

Nasrabadi, M. S., Sharafi, Y., & Tayari, M. (2016). A parallel grey wolf optimizer combined with opposition based learning. In *Swarm intelligence and evolutionary computation (CSIEC), 2016 1st conference on* (pp. 18–23). IEEE.

Natesan, G., & Chokkalingam, A. (2017). *Opposition Learning-Based Grey Wolf Optimizer algorithm for parallel machine scheduling in cloud environment*.

Ozsoydan, F. B. (2019b). *Effects of dominant wolves in grey wolf optimization algorithm*.

Pierezan, J., & Coelho, L. D. S. (2018). Coyote optimization algorithm: a new metaheuristic for global optimization problems. In *2018 IEEE congress on evolutionary computation (CEC)* (pp. 1–8). IEEE.

Pradhan, M., Roy, P. K., & Pal, T. (2017). Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system. *Ain Shams Engineering Journal*.

Punnathanam, V., & Kotecha, P. (2016). Yin-Yang-pair Optimization: A novel lightweight optimization algorithm. *Engineering Applications of Artificial Intelligence, 54*, 62–79.

Rahnamayan, S., Jesuthasan, J., Bourennani, F., Salehinejad, H., & Naterer, G. F. (2014). Computing opposition by involving entire population. In *Evolutionary computation (CEC), 2014 IEEE congress on* (pp. 1800–1807). IEEE.

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.

Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence, 80*, 20–34.

Shayanfar, H., & Gharehchopogh, F. S. (2018). Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Applied Soft Computing, 71*, 728–746.

Singh, P. K., Das, S., Sarkar, R., & Nasipuri, M. (2018). Feature Selection using harmony search for script identification from handwritten document images. *Journal of Intelligent Systems, 27*(3), 465–488. doi:10.1515/jisys-2016-0070.

Tang, D., Dong, S., Jiang, Y., Li, H., & Huang, Y. (2015). ITGO: Invasive tumor growth optimization algorithm. *Applied Soft Computing, 36*, 670–698.

Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. In *Computational intelligence for modelling, control and automation, 2005 and international conference on intelligent agents, web technologies and internet commerce, international conference on: 1* (pp. 695–701). IEEE.

Tsai, H. C., Tyan, Y. Y., Wu, Y. W., & Lin, Y. H. (2013). Gravitational particle swarm. *Applied Mathematics and Computation, 219*(17), 9106–9117. doi:10.1016/j.amc.2013.03.098.

Van Laarhoven, P. J., & A. E. , H. (1987). *Simulated annealing. In Simulated annealing: Theory and applications* (pp. 7–15). Dordrecht: Springer.

Webster, B., & Bernhard, P. J. (2003). *A local search optimization algorithm based on natural principles of gravitation*. Retrieved from https://repository.lib.fit.edu/handle/11141/117

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

Zhang, Q., Wang, R., Yang, J., Ding, K., Li, Y., & Hu, J. (2017). Collective decision optimization algorithm: A new heuristic optimization method. *Neurocomputing, 221*, 123–137.

Zhang, S., Luo, Q., & Zhou, Y. (2017). Hybrid Grey Wolf Optimizer using elite opposition-based learning strategy and simplex method. *International Journal of Computational Intelligence and Applications, 16*(02), 1750012. doi:10.1142/s1469026817500122.