

# Opposition-Based Multi-Tiered Grey Wolf Optimizer for Stochastic Global Optimization Paradigms

Vasudha Bahl, Banasthali Vidyapith, India\*

Anoop Bhola, Banasthali Vidyapith, India

## ABSTRACT

Researchers are increasingly using algorithms that are influenced by nature because of its ease and versatility. The key components of nature-inspired metaheuristic algorithms are investigated, involving divergence and adoption, investigation and utilization, and dissemination techniques. Grey wolf optimizer (GWO), a relatively recent algorithm influenced by the dominance structure and poaching department of grey wolves, is a very popular technique for solving realistic mechanical and optical technical challenges. Half of the recurrence in the GWO are committed to the exploration and the other half to exploitation, ignoring the importance of maintaining the correct equilibrium to ensure a precise estimate of the global optimum. To address this flaw, a multi-tiered GWO (MGWO) is formulated that further accomplishes an appropriate equivalence among exploration and exploitation, resulting in optimal algorithm efficiency. In comparison to familiar optimization methods, simulations relying on benchmark functions exhibit the efficacy, performance, and stabilization of MGWO.

## KEYWORDS

Grey Wolf Optimizer, Meta-Heuristic, Nature-Inspired Algorithm, Optimization, Swarm Intelligence

## INTRODUCTION

Meta-heuristic optimal techniques are becoming extremely prevalent in practical execution because they implement on basic concepts for easy execution, don't need gradient knowledge, could be employed on numerous problems spanning various fields (Shayanfar & Gharehchopogh, 2018; Gharehchopogh & Gholizadeh, 2019; Gharehchopogh et al., 2019; Abedi & Gharehchopogh, 2020; Majidpour & Gharehchopogh, 2018). Meta-heuristic algorithms have shown to be effective in understanding several stochastic and multimodal actual optimal problems. Chaotic and local hunt are also used in all Meta-heuristic implementations and for global optimum meta-heuristic algorithms may be useful (Khalandi & Gharehchopogh, 2018; Allahverdipour & Gharehchopogh, 2018).

Due to the ever-increasing complication of actual issues in engineering and technology, Global Optimization (GO) has evolved as indispensable for utmost optimization. The GO contains a large number of difficult multimodal optimization problems for which most conventional optimization lag or have unattainable investigations. Swarm Intelligence (SI) implementations are among the most

DOI: 10.4018/ijeoe.295982

\*Corresponding Author

This article published as an Open Access Article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

efficient and effective GO approaches. SI algorithms are computation frameworks that were created by mimicking natural seeking behavior patterns and are used to solve multi - objective optimization issues.

Populace dependent metaheuristic optimization is among the most robust strategies to overcome continuous and combinatorial optimization issues. Numerous real-time problems, on the other hand, are frequently framed as multi objective challenges with constrained resources (Sumpunsri et al., 2021). Two challenges arise when deploying nature-inspired methods to address Mining Algorithm with statistical properties: appropriately minimizing the ranges of the upper and lower limits of attributes, and effectively designing the assessment function. suggests utilizing a shrinking coefficient to minimize the interval of characteristics, while the assessment function is specified as a weighted average of support, reliability, participation, and shrinking coefficient (Fister et al., 2021).

GWO (Mirjalili et al., 2014) a competent and new onset meta-heuristic progressive optimization technique depend on the dominance and targeting behavior of grey wolves. Grey wolves (*Canis Lupus*), which belong to the Canidae folks, influenced the GWO procedure. Gray wolves live in clusters, with a batch size ranging from 5 to 12. The leader is known as alpha and is qualified to make decisions such as hunting, sleeping, and so on. The latter is known as beta, and it supports the alpha in making decisions. The alpha wolf should be appreciated by the beta wolf. Omega is the weakest grey wolf in terms of level, and it sends data to other wolves in the region. The rest of the grey wolves have given their names delta. Exploration and exploitation are two key divisions of metaheuristic approaches (Miandoab & Gharehchopogh, 2017). Exploration ensures that the algorithm reaches distinct suitable regions in the problem space, while exploitation ensures that global optimum is found in the specific area (Gharehchopogh et al., 2015). The usefulness and optimum results obtained metaheuristics in developing skew complex laminates in active workable situations is investigated in depth (Kalita, et al., 2021). In diverse-scale architecture, a conceptual regression performed via Genetic Programming (GP) paired with D-optimal layout is offered as an alternative theoretic foundation model for optimal algorithm (Kalita, Mukhopadhyay, et al., 2020). In (Kalita, Dey, et al., 2020) By integrating the high-accuracy of the structural analysis approach with the continuous enhancement potential of evolutionary methods, a high-precision structural optimal control model is established. The (Kalita, et al., 2018) study looks upon genetically optimized skew laminates, which have had their impact strength changed to enhance their elementary recurrence using an effective optimizer.

Grey wolves' distinctive prey technique and system approach influenced GWO. An improved GWO functionality is introduced in this study as GWO design is prone to falling into local optimum, particularly when utilized with high- proportion facts. The discovery function of GWO is improved, and the deficiency of GWO is compensated, by incorporating the global-search potential of GWO into MGWO to upgrade its strongest three solutions, which are alpha, beta, and delta wolf. The suggested methodology has global-search potential, and it could avoid falling into the local optimum and jumping out of the local optimum in elevated populations, according to preliminary experimental study and thus compared with Particle Swarm Optimization-GWO (PSOGWO), GWO-Cuckoo Search (GWOCS), Enhanced-GWO (EGWO), Augmented- GWO (AGWO), Particle Swarm Optimization (PSO) and GWO algorithms.

The paper is arranged as: Section 2 includes a summary of the GWO algorithm. Section 3: Provides a review of the literature. Section 4 contains a framework of the research MGWO algorithm. Section 5: The MGWO algorithm is validated on confined benchmark functions, followed by an experimental investigation and analysis of the results. Section 6: The work's interpretation is offered, as well as its future direction.

## **GREY WOLF OPTIMIZER (GWO)**

GWO is a common SI algorithm that is influenced from the hierarchical administration and chasing system of grey wolves. They are considered prime target, and require a community density of 5–12 individuals. In GWO, alpha ( $\alpha$ ) is viewed as the supreme overwhelming portion amidst the pack. In

figure 1, the subsidiary to are beta ( $\beta$ ) and delta ( $\delta$ ), which aids to regulate most of wolves classified as omega ( $\omega$ ) appeared in figure 1. The  $\omega$  wolves are of the most minimal positioning in the chain. The alphas have the most elevated level in the pecking order and omega the least. The alphas are the most grounded in the pack and provide the administration to the gathering. These wolves can distinguish the area of prey and the henceforth entire pack will proceed and attack.

The phases of trapping are (Mirjalili et al., 2014):

1. Locating, pursuing, and arriving at prey.
2. Encompassing and annoying prey till it stops.
3. Attacks the prey.

The numerical representation of the wolves for prey chasing and attacking is presented as below:

1. **Encompassing Prey:** Wolves enclose the target in the chase obtained numerically as Eq 1, 2:

$$\vec{D} = \left| \vec{C} * \vec{X}_p(t) - \vec{X}(t) \right| \tag{1}$$

$$\vec{X}(p+1) = \vec{X}_p(t) - \vec{A} * \vec{D} \tag{2}$$

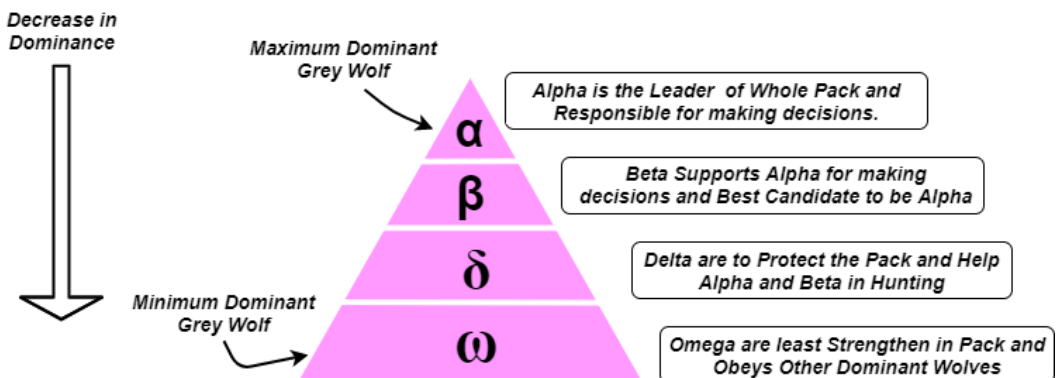
t shows present emphasis,  $\vec{A}$  and  $\vec{C}$  are coefficient trajectory,  $\vec{X}_p$  is placement trajectory of target, and  $\vec{X}$  demonstrates placement trajectory of grey wolf.  $\vec{A}$  and  $\vec{C}$  are determined in eq 3,4,5:

$$\vec{A} = 2a * \vec{r}_1 - a \tag{3}$$

$$\vec{C} = 2\vec{r}_2 \tag{4}$$

$$a = 2 \left( 1 - \frac{t}{T} \right) \tag{5}$$

Figure 1. Dominance of Grey wolves



where segments of  $\vec{a}$  straightaway diminished along 2 to 0 throughout emphases and r1, r2 are arbitrary trajectory in [0, 1].

2. **Tracking:** Grey wolves can sense where they are being hunted and will surround them. The  $\alpha$  is usually in charge of the pursuit. The  $\beta$  and  $\delta$  can occasionally engage in pursuing as well. Nonetheless, in a theoretical hunt space, Developers also had no knowledge where the perfect area is (prey). To numerically recreate the pursuit behavior patterns of wolves, Designers believe that  $\alpha$  (finest competitors' scenario),  $\beta$  and  $\delta$  have broader insight about the possible prey zone. Subsequently, they keep the first 3 top arrangements made thus far and require the other chase agents (adding up  $\omega$ ) to reposition themselves in accordance with the placement of the top candidate solutions seen in Figure 2. The foremost necessary category  $\alpha$ ,  $\beta$  and  $\delta$  lead to the strongest test methods in terms of strength. Then, using the calculation below, the wolf locations with these three categories are revised in eq 6,7,8:

$$\vec{D}_\alpha = \left| \vec{C}_1 * \vec{X}_\alpha - \vec{X} \right| \quad (6)$$

$$\vec{D}_\beta = \left| \vec{C}_2 * \vec{X}_\beta - \vec{X} \right| \quad (7)$$

$$\vec{D}_\delta = \left| \vec{C}_3 * \vec{X}_\delta - \vec{X} \right| \quad (8)$$

$\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$  is the placement trajectory of  $\alpha, \beta$  and  $\delta$  respectively. Within standard conditions, first three phases of grey wolf are believed to be familiar with the following location of the target on the chase. While acquiring abovementioned conditional probability, the wolves will formulate approximation to the acquired location to achieve the entire modification in eq 9,10,11:

$$\vec{X}_1 = \vec{X}_\alpha - A_1 * \vec{D}_\alpha \quad (9)$$

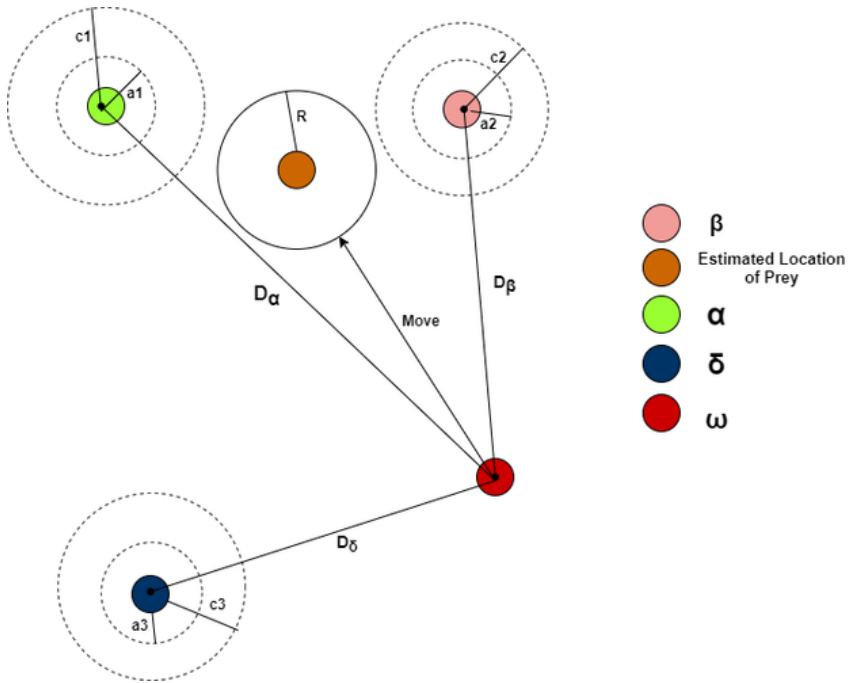
$$\vec{X}_2 = \vec{X}_\beta - A_2 * \vec{D}_\beta \quad (10)$$

$$\vec{X}_3 = \vec{X}_\delta - A_3 * \vec{D}_\delta \quad (11)$$

$\vec{X}$  demonstrates placement vector of a grey wolf where t shows recent emphasis given by eq 12:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (12)$$

Figure 2. Location update in GWO



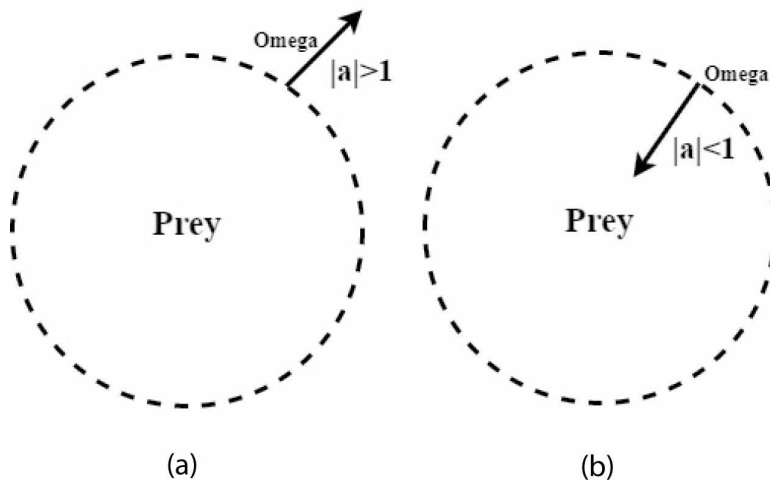
### Attacking Prey (Exploitation)

As referenced grey wolves conclude pursue by striking the target as it halts motion. To numerically demonstrate motion to the target value of  $\vec{a}$  is declined. The change scope of  $\vec{A}$  is likewise diminished by  $\vec{a}$ .  $\vec{A}$  is irregular quantity in the span  $[-a, a]$ ,  $a$  is diminished straightaway from 2 to 0 throughout the reduplications. At the point when arbitrary quantity of  $\vec{A}$  lies in  $[-1, 1]$ , the following location of a hunt specialist can be in any role amidst its present range and location of target.  $|A| < 1$  powers the wolves to assault target. Subsequently the assault again they look for target in the following cycle, accordingly, they once more track down the following best finding  $\alpha$  amidst every wolf. This action replicate till the termination benchmark is accomplished.

### Discover Target (Exploration)

Grey wolves hunt primarily using  $\alpha$ ,  $\beta$  and  $\delta$  locations. They separate to hunt for prey and then combine to strike prey. To numerically display  $\vec{A}$  is used with unsymmetrical values more distinguished  $< 1$  or  $< -1$  to force the hunt maestro to detached from target. This encourages discovery and enables GWO to examine the globe. Figure 3 also reveals that when  $|A| > 1$  the grey wolves are compelled to detached from target to track down a fitter prey.  $\vec{C}$  is another aspect of GWO that encourages exploration. The  $\vec{C}$  vector, as shown in Eq. 4, includes arbitrary values in the range  $[0, 2]$ . This portion gives arbitrary loads to prey to stochastically prioritize ( $C > 1$ ) or diminish ( $C < 1$ ) the impact of target in characterizing the range. This allows GWO to behave in irregular manner during optimization, preferring exploration and averting local optima. It's worth noting that, in comparison to  $A$ ,  $C$  does not diminish linearly. We need  $C$  to give arbitrary values consistently to accentuate exploration not only during the starting cycles but also during the final cycles.

Figure 3. (a) Discovering Target (b) Attacking Prey



The discovery phase begins with the GWO algorithm generating an arbitrary populace of grey wolves.  $\alpha$ ,  $\beta$ , and  $\delta$  wolves predict the approximate whereabouts of the target over the throughout iterations. The distance between each finding and target is modified. To affirm discovery and exploitation, the criterion  $a$  is reduced from 2 to 0. As  $|A| > 1$  candidate findings appear to separate target and join the target as  $|A| < 1$ . At last, the GWO is terminated when an end condition is met.

In (Kumar, et al., 2021), a new adaptive GP-GWO strategy for biodiesel procedure optimization is suggested, in which kinematic viscosity is demonstrated as a symbolic regression process model to address the impact on optimized process parameters using a GWO optimal solution.

### Limitations

Low solution accuracy, sluggish convergence, and poor local searching capabilities are all drawbacks of the typical GWO algorithm. The MGWO algorithm is presented to tackle limited optimization issues in order to overcome the shortcomings of GWO. Since GWO variations, including MGWO, rely on the arithmetic averaging formula for individual location upgrade, our primary goal in MGWO is to investigate the algorithm's behavior by combining the arithmetic averaging formula with the fitness factor.

### Motivation

The wolves' social conduct and levelled demeanor are prominent and readily noticeable (Figure 1). This conduct prompts a productive chasing strategy. This social insight of the wolves, the strength of the pioneer wolf that is alpha and other i.e., beta, delta, and omega alongside the adaptable system of searching, drawing closer lastly chasing the prey are the three fundamental rousing components behind the powerful working of the GWO calculation. It is unsuitable for more complex functions, and it may still be susceptible to becoming stuck in local optima. To counter this shortcoming and boost its exploration ability. Multi-layered GWO (MGWO) a variant dependent on GWO calculation is developed to answer relevant genuine issues. The fundamental inspiration driving MGWO is to improve the slow convergence rate and inadequate solution space capability.

### BACKGROUND

Integration on an automated platform cluster would be a popular mode for completing necessary activities. Community decision-making is a crucial technology for facilitating unmanned cluster

collective work due to the complexity and variety of the domain and task categories. Gray wolves have excellent intellectual and coordination abilities, and can trace and envelop prey in a dynamic world easily. Knowledge awareness, division of work, and collaboration are intelligent action attributes that are compatible with the judgement requirements of an automated group. GWO is a modern SI optimization algorithm that is well suited to resolving such critical decision optimization problems.

## Related Works

Focusing on the inadequacies of conventional GWO as sluggish convergence speed and effective plunge to search space, an enhanced grey wolf optimization (EGWO) (Joshi & Arora, 2017) is applied to disintegrate the collective hunting behaviour of wolves in phases. Different mathematical strategies are used to refine the theoretical formulation built from discovery and monitor, chase, encirclement, and strike predators. The process phase transitioning model is designed to interact with structure mutations in actual environments, allowing the cluster to solve problems and respond faster. To resolve limitation of GWO, EGWO proposes a stronger hunting strategy that emphasizes on a balanced combination between discovery and exploitation, resulting in optimal method efficiency.

For superior targeting efficiency, a fast and structured Augmentation GWO (AGWO) (Qais et al., 2018) technique is suggested. By altering the conduct of the control parameter ( $a$ ) and location modification, the AGWO aims on expanding the probability of the exploration over exploitation. The AGWO is well suited to applications with a small count of search agents like the power grid. A novel hybrid method relies on GWO and cuckoo search (GWOCS) is introduced to retrieve the specifications of various PV cell models from measured evidence under various conditions. New resistance investigation technique for selection level populations ( $\alpha$ ,  $\beta$ , and  $\delta$ ) is formulated in GWOCS (Long et al., 2020) to increase GWO heterogeneity. The potential of GWOCS to strike a balance between discovery and exploitation is its key benefit. Particle swarm optimization (PSO) (Wang et al., 2018) is a community stochastic optimizer inspired by the sophisticated active participation of certain entities such as bird flocks or fish schools. Researchers have created updated models of the strategy to meet higher needs, introduced innovative implementations in a variety of fields, conducted conceptual evaluation of the impact of key variables, and suggested numerous variants of the approach. The PSO is based on (Kennedy & Eberhart, 1995) algorithm, with variations recommended by (Mezura-Montes et al., 2011; Pedersen, 2010). The preliminary particles are created and given basic flow rates in the PSO. It makes an ideal (lowest) fitness values and the right place by evaluating the objective function at every entity site. It determines further velocities depending on the present velocity, the best locations of the objects individually, and their surroundings. It then repetitively upgrades particle positions, velocities, and surroundings (the new position is the previous one plus the velocity, updated to hold objects in scope). Repetitions continue until the process meets a threshold for terminating. Particle swarm optimization's (PSO) exploitation potential is combined with the grey wolf optimizer's (GWO) exploration strength in this combination algorithm (Şenel et al., 2019). By swapping a particle of the PSO with a minimal chance with a particle moderately strengthened with the GWO, the approach incorporates two approaches. The method is tested against five separate benchmark functions as well as three real concerns, including estimation methods for flow sheeting, and leather nesting (LNP) so on. Approach also conforms to more efficient outcomes with less iterations, according to the findings.

Hybrid implementations for the optimizing of multifaceted complex domain, FWGWO is introduced, which is focused on GWO and Fireworks Algorithm (FWA) (Yue et al., 2020). The FWGWO algorithm blends GWO's strong exploitation ability with the FWA's strong exploration potential. In terms of balancing exploitation and discovery, this algorithm employs an appropriate equilibrium coefficient. The equilibrium multiplier determines the likelihood of exploitation or discovery. The FWGWO algorithm has a quick convergence speed and can resist the local optimum as much as feasible by adjusting the equilibrium multiplier. Numerous uncertainties are common in sophisticated manufacturing facilities. As a result, advanced fuzzy optimization (metaheuristics)

approaches have become mainstream, and are now required for efficient design, development, and services. However, when compared to the large problems, such cutting-edge solutions have numerous limitations. The fuzzy random matrix theory (RMT) is proposed as an update to the cuckoo search (CS) solution to address the fuzzy large-scale multi-objective (MO) optimization issue in order to enhance the effectiveness of metaheuristics among those (GANESAN et al., 2020).

The Binary Grey Wolf Optimizer (BGWO) (Pei et al., 2020) adds the GWO's functionality to binary optimization problems. A parameter affects analytic discovery and exploitation in the BGWO location modification equations by controlling the variables of A and D. This investigates the scope of AD attributes in binary conditions and suggests a new modifying equation for a variable that give equilibrium to global and local search abilities. The upgraded GWO based on tracking mode (TGWO), focused on searching (SGWO), the modified GWO based on monitoring and seeking (TSGWO) is presented in (GUO et al., 2020). It is used to solve the simplest pressure vessel design engineering challenge. The suggested methodology has better efficiency in terms of enhancing the ideal, and it has benefits of exploration, according to results obtained.

I-GWO (Mohammad et al., 2021) suggested a method to overcome the problem of population size, the disparity among exploitation and discovery, and the GWO algorithm's premature convergence. The I-GWO algorithm takes advantage of a modern mobility technique known as dimension learning-based hunting (DLH) sensitive approach, which was inspired by wolf hunting activity in the wild. DLH uses a unique approach to creating a community for each wolf, one in which neighbours' knowledge can be exchanged. The DLH analysis method utilizes dimension learning to improve the equilibrium among local and global quest while maintaining versatility. IGWO (Li et al., 2021) is a method that allows engineers solve design problems. The initial position of the grey wolf population is generated using a tent map, which equalises wolf populace and provides the basis for a manifold universal quest behaviour. Second, to prevent the method crashing into local optima, Gaussian mutation distortion execute different functionality on existing optimum. Finally, a cosine control factor is added to align the algorithm's global and local exploration proficiency and accelerate convergence. The IGWO algorithm is used to solve four engineering optimization problems with varying degrees of complexity. Investigators have intensified their usage meta-heuristic strategies in the fields of industry, science, and technology as a result of their growth. The K-means clustering is applied to improve the existing GWO's performance; the updated algorithm is known as K-means clustering GWO (KMGWO) (Mohammed et al., 2021). Cat swarm optimization (CSO), whale optimization algorithm-bat algorithm (WOA-BAT), WOA, and GWO are all contrasted with KMGWO for investigations. In addition, hybrid Particle Swarm Optimization– GWO (PSO–GWO) (Prithi & Sumathi, 2021) accurately consume power and safely send information in an extended direction. To incorporate the changing aspect of the system, Learning Dynamic Deterministic Finite Automata (LD<sup>2</sup>FA) is implemented. The main purpose of LD<sup>2</sup>FA is to give experienced and authorized string to PSO–GWO in order to optimise the routes. To ensure the best route, PSO–GWO is followed to select best subsequent node for each direction. Expanded GWO has major three wolves are alpha, beta, and delta, just like they are in GWO. The following wolves, on the other hand, choose and change location dependent on prior and first three wolves (Seyyedabbasi & Kiani, 2021). The incremental GWO is another suggested method that is dependent on the incremental model. This system upgrades each wolf's location based on the positions of all the wolves chosen before it. There's a chance that this algorithm can find solutions (hunts) faster than other implementations in the same group. However, since they rely on one another, they cannot always be able to come up with a good alternative. Both algorithms rely on exploration and discovery.

Each of the aforementioned optimization algorithms must resolve the exploration and exploitation of a local optimum. An optimization algorithm must develop an equilibrium between exploration and exploitation in order to be optimal. The Multi-tiered GWO (MGWO) is suggested in this paper to resolve the discovery and exploitation trade-off in the existing GWO. Over the iteration process, various tasks with different orientations are used to determine the stability of the GWO algorithm



for varying exploration and exploitation formulations. Increasing discovery in relation to exploitation speeds up convergence and prevents the locking impact of local minima. Numerous performance aspects, such as precision, authenticity, concurrence, and enumeration, are tested using 23 benchmarks to thoroughly assess the efficacy of MGWO.

## MULTI-TIERED GREY WOLF OPTIMIZER (MGWO)

### Discussion

This new variant aims to increase GWO's global exploring capabilities and computational complexity by redesigning the encircling framework and location reform equations. MGWO begins with creating a populace of P grey wolves  $P_j, j = 1, 2, \dots, n$  actually like GWO, where n is issue measurement. All through this work, the grey wolves will be referred to as search agents. The populace generated based on fitness objective function and is arranged in order. The fittest (with a base target esteem) populace part is known as  $X'_\alpha$ , the second-best worth is  $X'_\beta$ , the third best is  $X'_\delta$ , and the excess populace individuals are signified as  $X'_\omega$ . The algorithmic model of MGWO is introduced in Algorithm 1. The analytical, quantitative, convergence capability of MGWO have been evaluated on 23 benchmark functions and are contrasted with current metaheuristics.

### Mathematical Formulation of Proposed Algorithm

The traditional GWO algorithm tries to find for a global solution by imitating grey wolves' prey tracking process. The location modification formula, that is measured as the average of three strongest grey wolf positions, is at the core of the GWO algorithm. This method performs well for simple problems, but it fails to provide accurate estimates for complex multimodal issues with higher dimensions. As a result, alternatives to complex multimodal global optimization problems demonstrate premature convergence and low quality. We suggest the MGWO to enhance the efficiency of the standard GWO. The location optimization formula in MGWO algorithm is dependent on the objective fitness function  $f_k$ . The leader agents as  $X'_\alpha$ ,  $X'_\beta$ , and  $\overrightarrow{X'_\delta}$  are selected as according to the best three fitness value of objective fitness function  $P(f_k)$  by applying sorting. Estimation is used to determine the location of the target.

The top three ranks,  $\alpha'$ ,  $\beta'$  and  $\delta'$ , lead to the three results with optimal condition. Thus, using the calculation below, the wolf locations with all these three top ranks are modified in eq 13,14,15:

$$\overrightarrow{D'_\alpha} = \left| \overrightarrow{C_1} * \overrightarrow{X'_\alpha} - \overrightarrow{X} \right| \quad (13)$$

$$\overrightarrow{D'_\beta} = \left| \overrightarrow{C_2} * \overrightarrow{X'_\beta} - \overrightarrow{X} \right| \quad (14)$$

$$\overrightarrow{D'_\delta} = \left| \overrightarrow{C_3} * \overrightarrow{X'_\delta} - \overrightarrow{X} \right| \quad (15)$$

where,  $\overrightarrow{X'_\alpha}$ ,  $\overrightarrow{X'_\beta}$ ,  $\overrightarrow{X'_\delta}$  placement trajectory of  $\alpha'$ ,  $\beta'$  and  $\delta'$  respectively.

Within standard conditions, first three stages of grey wolves are believed to be familiar with the following locations of the predators on the quest. The wolves will conduct the final version based

on the acquired location using the equation below after retrieving abovementioned fitness value in eq 16,17,18:

$$\vec{X}'_1 = \vec{X}'_\alpha - A_1 * D_\alpha' \quad (16)$$

$$\vec{X}'_2 = \vec{X}'_\beta - A_2 * D_\beta' \quad (17)$$

$$\vec{X}'_3 = \vec{X}'_\delta - A_3 * D_\delta' \quad (18)$$

$\vec{X}'_t$  demonstrates the placement trajectory of a grey wolf where t shows the present emphasis. As a result, via equation, define the closest agent to be identified as the new best option given by eq 19:

$$\vec{X}'(t+1) = \frac{\vec{X}'_1 + \vec{X}'_2 + \vec{X}'_3}{3} \quad (19)$$

As shown in Eq 3, 4,5 the modified coordinates are determined using coefficient vectors  $\vec{A}$  and  $\vec{C}$ .

### Parameters for Objective Functions

1. **Coverage Area (f1):** The coverage area of each wolf, where a proper distribution of search agents should retrieve both similar coverage area across search agents and coverage for every wolf, as in eq 20:

$$A_{Cover} = (d_{far}^2 * \pi) / n \quad (20)$$

$A_{Cover}$  is the coverage area and  $d_{far}^2$  is the squared interspace to the extreme wolf from the median. Hence, all wolf is within  $d_{far}^2 \pi$ , the ring-shaped area from median of the wolf to  $d_{far}^2$ , and any wolf interior of ring enclosed by a search agent has interspace lessened than the extent to that search agents. Further, correlating interspace is manageable than demonstrating if a wolf is interior of ring as in eq 21:

$$R_{Cover} = \sqrt{\frac{A_{Cover}}{\pi}} = \sqrt{\frac{d_{far}^2 * \pi}{n * \pi}} = \frac{d_{far}}{\sqrt{n}} \quad (21)$$

where  $R_{Cover}$  is the extent to regulate whether a wolf is inside scope  $A_{Cover}$ . The group of wolfs enclosed by the k-th search agent is indicated as:

$$coverK = \{search\ agentID \mid Distance(search\ agentID, mk) < R_{Cover}\}, k = (1, 2, 3 \dots n) \forall ID$$

where Distance ( $search\_agent_{ID}, mk$ ) is the range from wolf particularly characterized by ID to the k-th search agent. Observe  $R_{Cover}$  examine that the total scope area is split uniformly. Therefore, additional wolfs enclosed by the search agents signify a finer organization. Accordingly, a fitness estimate can be indicated as eq 22:

$$f1 = \left| \bigcup_{k=1}^N cover_K \right| \quad (22)$$

Here  $||$  symbolizes cardinality of group (i.e., element count) and union avert total overlying wolfs enclosed by numerous best search agents.

2. **Balancing Element (f2):** There is necessity of equilibrium in group of search agents. For sake of arbitrary formation of agents, there is a probability that several large groups are set up and several tiny groups of search agents. So, this factor is requisite to stabilize the dissipation of vitality in eq 23:

$$f2 = \sum_{k=1}^m \left( \frac{n}{m} - l_k \right) \quad (23)$$

where  $n$  is overall count of search agents,  $m$  is Total aggregate of best search agents,  $l_k$  is numbers of wolfs in the group  $k$ . Alternatively, individually reduce one and all fitness function it is preferable to shrink the union of the above fitness function as expressed in mathematical statement. The above fitness functions are energetically in consistency of both.

If best search agents are in range, then wolfs will consider as optimal set of search agents, that implies  $search\_agents(i), d_{toBest} < R$  then compute the Fitness function ( $f_i$ ) as:

$$f_i = \mu * f1 + \sigma * f2$$

where  $\mu$  and  $\sigma$  represent constant variable and  $\mu + \sigma = 1$  and if ( $search\_agents(i), d_{toBest} < R$ )

Compute eq 24:

$$f_k = \mu * \left| \bigcup_{k=1}^N cover_K \right| + \sigma * \sum_{k=1}^m \left( \frac{n}{m} - l_k \right) \quad (24)$$

The objective for deciding GWO atop meta-heuristic techniques is that GWO has rapid convergence rate. Additionally, GWO dominance uninterrupted depletion of search space and selection parameters are fewer. Moreover, avert local optima.

The position optimization calculation is changed to represent the current measured objective fitness,  $f_k$  given in eq 25:

$$f_k = \mu * \left| \bigcup_{k=1}^{Np} cover_K \right| + \sigma * \sum_{k=1}^{Np} \left( \frac{n}{m} - l_j \right) \quad (25)$$

where  $N_p$  is Total Populace Proportions.

This system is highly useful in a dynamic environment where the issue landscape has narrower and numerous ranges. MGWO is deployed on tracking and seeking mode.

**Step 1:** Instantiate MGWO benchmark framework: Populace Proportions ( $N_p$ ), maximal recurrence count ( $Max\_N_p$ ), Remembrance Bank (RB), Replace Count (CDC), Arbitrary Initial Rate, Arbitrary placement of  $\alpha'$ ,  $\beta'$  and  $\delta'$  wolf ( $\vec{X}'_\alpha$ ;  $\vec{X}'_\beta$ ;  $\vec{X}'_\delta$ ).

**Step 2:** Amend orientation of  $\alpha$ ,  $\beta$  and  $\delta$  applying Eq. 16, 17, 18. Enumerate the fitness value every wolf. Evaluate  $X'_\alpha$  as finest quest agent, and assess fitness value  $f(X'_\alpha)$  using eq 25.

**Step 3:** The MGWO algorithm updates the placements of  $\alpha$ ,  $\beta$  and  $\delta$  using Eqs. 16, 17, 18. MGWO approach employs the seeking mode to amend position of  $\alpha$  and alerts  $\beta$  and  $\delta$  with tracking mode by using Eqs. 16, 17, 18; MGWO implement seeking feature to change the locations of  $\alpha$ ,  $\beta$  and  $\delta$  to create a set of entities to occupy the quest storage database, duplicate the entities in the existing seeking phase: Establishing  $RB=5$ ,  $X'_1$ ,  $X'_2$ ,  $X'_3$  are freed in memory cache as nominees. Eq. 25 repeats the candidate solution  $N_p$  times,  $N_p = RB - 1$ ; fitness values of all nominee sites in the recall cache are determined individually at this stage.

**Step 4:** Upgrade  $X$  of applying Eq. 19 amend the placement  $\vec{X}'$  of each wolf applying Eq. 16,17,18.

**Step 5:** In case  $f(X') < f(X)$ , then  $f(X') = f(X)$ , else constant.

$$\text{In case } f(X) < f(X'_\alpha), \text{ then } f(X'_\alpha) = f(X)$$

**Step 6:** Termination criteria met and return optimal outcome  $f(X'_\alpha)$ , else reinstate Step 2.

The flow chart of MGWO is given in Figure 4 and Pseudo-code MGWO deployed on chasing and hunting process is outlined Considering arbitrary initialization of grey wolf populace is given in Algorithm 1.

## Advantages and Disadvantages of the Proposed Methodology

The location, speed, and convergence precision of the  $X'_\alpha$  agent have been strengthened in this version by using MGWO's location modification in Eq. (19) for stability among the exploration and the exploitation procedure and lengthening the convergence overall performance of GWO. In addition, the chaotic initialization strategy is used to construct the preliminary populace, which speeds up the overall computational efficiency of the GWO algorithm.

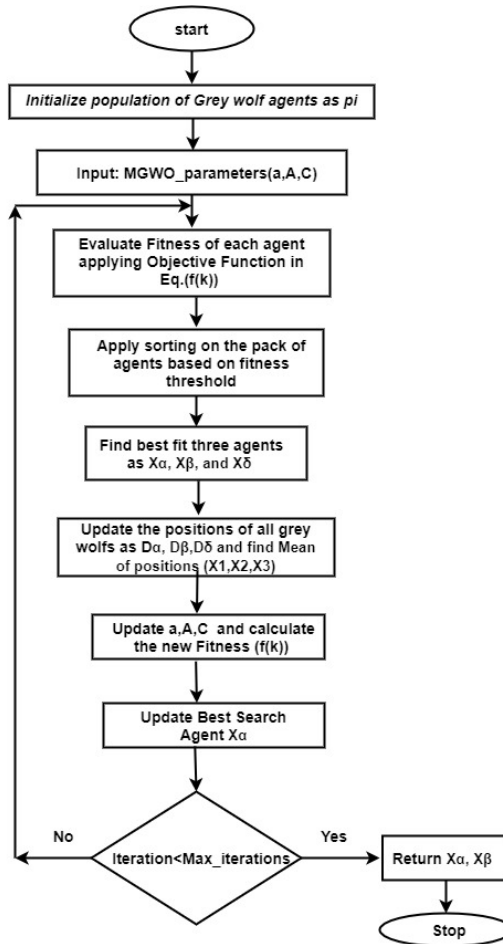
The investigational findings of 23 typical evaluation metrics show that the proposed MGWO algorithm outperforms the conventional GWO in terms of optimization accuracy, robustness, and optimal execution. However, the MGWO algorithm is not without flaws. F5, F19, F20 have functionality problems, MGWO and GWO perform in same manner.

## SIMULATION EXPERIMENTS AND RESULT ANALYSIS

### Benchmark Functions

They are standardized mechanisms derived from natural science research. These are normally complex and impartial, making analytical expressions impossible to solve. Benchmark functions have long been an important tool for evaluating the accuracy, performance, and validity of optimization algorithms. They ranged from the range of uncertain peaks in the function, reusability and dimensionality. The benchmark functions are being categorized quantitatively using the five essential characteristics (Jamil & Yang, 2005):

Figure 4. Flowchart of Optimization of MGWO



1. Continuous or uncontinuous
2. Differentiable or nondifferentiable
3. Separable or nonseparable
4. Scalable or nonscalable
5. Unimodal or multimodal

A total of 175 benchmark functions are interpreted within the context. We selected 23 benchmark functions, ranging in complexities from basic to advanced, that included in this paper. They are ideal for evaluating the capabilities of the algorithms and they are all scalable. The equations are all n-dimensional, and the domain limits the source vectors  $y = (y_1, y_2, \dots, y_n)$ . The domain's maximum and minimum values are  $ub$  and  $lb$ , respectively. The single result variables are all zeros technically for ease.

To assess the efficiency of the suggested technique from various viewpoints, three classes of test functions are used: unimodal (F1-F7, F18-F19), multimodal (F8-F13, F20-F21), and fixed-dimension multimodal functions (F14-F17) (Liang et al., 2005; Awad et al., 2017).

### Assumptions

MATLAB is used to support this research. Table 1 shows specification of parameters.

**Algorithm 1. Pseudo Code: MGWO Algorithm**

<p><b>Procedure: Initialize MGWO</b> (<math>a, A, C, pi</math>)</p>
<p><b>Notation List:</b></p> <ol style="list-style-type: none"> <li>1. <math>P_i</math> = Populace Proportions</li> <li>2. <math>Np</math> = Total Populace Proportions</li> <li>3. <math>\vec{A}</math> and <math>\vec{C}</math> = Coefficient Vector</li> <li>4. <math>Max\_Np</math> = Threshold Count in RB</li> <li>5. Placement of <math>\alpha'</math>, <math>\beta'</math> and <math>\delta'</math>, as <math>(X'_\alpha; X'_\beta; X'_\delta)</math></li> <li>6. <math>X'_\alpha</math>, as the best search agent</li> <li>7. <math>n</math> = aggregate of search agents</li> <li>8. <math>m</math> = Sum of best search agents</li> <li>9. <math>l_k</math> = numbers of wolfs in the group <math>k</math>.</li> <li>10. <math>\mu, \sigma</math> = constant variable and <math>\mu + \sigma = 1</math></li> </ol>
<p><b>Pseudocode: Algorithm BEGIN</b></p> <p>// Initialization Phase</p> <ol style="list-style-type: none"> <li>1. Initialize packs GWO <math>P_i</math> as {p1, p2, p3, p4, p5,...}, <math>RB = 5</math></li> <li>2. <math>Np = Count(P_i)</math></li> <li>3. Initialize <math>a, \vec{A}</math> and <math>\vec{C}</math></li> </ol> $\vec{A} = 2a * r_1 - a, \vec{C} = 2r_2$ $a = 2 \left( 1 - \frac{t}{T} \right)$ <p><math>a</math> linearly diminished from 2 to 0 over repetition and <math>r_1, r_2</math> arbitrary vectors in [0, 1].</p> <ol style="list-style-type: none"> <li>4. <math>X_p</math>: placement of prey, <math>X(t)</math> is search agent(wolf), <math>x(t+1)</math> is current fitness, on the basis of this current fitness we find <math>X'_\alpha</math> = best, <math>X'_\beta</math> = second-best, <math>X'_\delta</math> = third- best search agent</li> </ol> $\vec{D} = \left  \vec{C} * X'_\alpha(t) - X(t) \right $ $X(t+1) = X_p(t) - \vec{A} * \vec{D}$ <p>// Working Phase</p> <ol style="list-style-type: none"> <li>5. For <math>k = 1</math> to <math>Max\_Np</math> do</li> <li>6. Calculate Fitness () Using objective function as in equation</li> <li>7.</li> <li>8. <math>P(x) = (Sort(Fitness(P(f_i))), 1 &lt; x \leq Np)</math></li> <li>9. EndFor</li> <li>10. Select the leader agents as <math>X'_\alpha, X'_\beta,</math> and <math>X'_\delta</math> according to the best three fitness value of <math>P(f_i)</math></li> <li>11. Modify placement of prey applying equation</li> </ol> $\vec{D}_\alpha = \left  \vec{C}_1 * X'_\alpha - \vec{X} \right , \vec{D}_\beta = \left  \vec{C}_2 * X'_\beta - \vec{X} \right , \vec{D}_\delta = \left  \vec{C}_3 * X'_\delta - \vec{X} \right $ $X'_\alpha = X'_\alpha - \vec{A}_1 * \vec{D}_\alpha, X'_\beta = X'_\beta - \vec{A}_2 * \vec{D}_\beta, X'_\delta = X'_\delta - \vec{A}_3 * \vec{D}_\delta$ <ol style="list-style-type: none"> <li>12. Update position for the agent as</li> </ol> $\vec{X}^-, (k+1) = \frac{X'_\alpha + X'_\beta + X'_\delta}{3}$ <ol style="list-style-type: none"> <li>13. Find the nearest agent to be selected as new best search</li> <li>14. EndFor</li> <li>15. Amend <math>a, \vec{A}</math> and <math>\vec{C}</math> using step 3</li> <li>16. Enumerate Fitness (<math>f_i</math>) of all search agents using step 6</li> <li>17. Modify Location of present search agent <math>X'_\alpha, X'_\beta, X'_\delta</math></li> <li>18. <math>k=k+1</math></li> <li>19. return <math>X'_\alpha</math></li> <li>20. Best_Agent ← <math>X'_\alpha</math></li> <li>21. Second_Best_Agent ← <math>X'_\beta</math></li> <li>22. Endfor</li> </ol> <p><b>END Algorithm</b></p>

**Simulations and Performance Analysis**

With a Populace Proportions of 30, all of the algorithms are performed for 500 repetitions. As demonstrated in Table 1, the efficacy of methods is evaluated on known benchmark functions to analyse the mean and standard deviation of all methods. The maximum iterations count is set as the terminating condition for all algorithms. PSOGWO, GWO, GWOCS, EGWO, AGWO, and PSO are contrasted to the potential surveillance and targeting mode-based MGWO. The modified simulation is conducted ten times for each benchmark function, beginning with different populations that were randomly generated. Table 1 shows standard deviation of the closest estimated outcome in the previous iteration.

The standard deviation can be used to analysed the computation system efficiency. The convergence curve is a visual representation of the algorithm’s optimization results. The convergence curves of unimodal and multimodal functions (F1-F7, F18-F19), multimodal (F8-F13, F20-F21), and fixed-dimension unimodal (F1-F7), multimodal (F8-F13, F20-F21), and fixed-dimension unimodal (F8-F13, F20-F21), and fixed-dimension multimodal functions (F14-F17), are shown in Figure 5 (a) – (u).

**Table 1. Parameter Specification**

Parameter	Specification
Populace Proportions	30
Maximum count of Iterations	500
Total count of runs	30
constant variable, $\mu, \sigma$	$\mu + \sigma = 1$
numbers of wolfs in the group $k$	30

Simulations demonstrate that the proposed enhanced algorithm outperform other algorithms. The MGWO outperforms other strategies on the most of unimodal benchmark functions, as shown by the outcomes of the implementations on the unimodal test functions in Table 1 and the convergence curves of the unimodal functions, Figure 5 (a)-(g) and Figure 5(p)-(q). Table 2 shows that the MGWO method's accurate approximation for optimizing unimodal functions F1-F4, F7, F18, and F19 is the nearest to the optimal solution; convergence towards optimum occurs only in the ultimate implementations, as shown in F5 and F6, but average weight achieved by MGWO is the nearest to the optimized predicted weight, and To summarize, the results demonstrate the effectiveness MGWO implementations outperform well-known meta-heuristics in resolving complex benchmark functions.

In contrast to certain other meta-heuristic strategies, MGWO has a faster convergence rate. MGWO performs well, particularly in the benchmark functions F1, F2, F3, F4, F5, F6, F8, F10, and F12. Similarly, MGWO has respectable functionality to PSOGWO, GWO, GWOCs, EGWO, AGWO, and PSO for F11, F16, F18, F19, and F20. As can be seen from the chart, GWOCs and MGWO both determined the optimum “zero” for F11. As indicated in Table 2, MGWO outperformed on seven of the nine test tasks. It's possible that MGWO's disappointing leadership on F9 and F22 is related to the structural properties of such problems, which each have many local minimum options. These issues are not simple to solve. Because the suggested method's assessment stage in the hunt strategy performed in a balanced manner, the presented heuristics demonstrate that it has a significant dominance over GWO in the hunt operation. The recommended MGWO methodology has a high rate of convergence.

## FUTURE RESEARCH DIRECTIONS

As a result, the algorithm discussed in this work may be beneficial to researchers in a variety of situations. The MGWO, on either hand, can be used for routing and localization issues due to its stable behavior and ability to completely encircle the aim. The usage of these strategies for node localization and discovering route in sensor networks, as well as to interpret reform fitness functions in computing weight codes in artificial neural networks and deep learning-dependent frameworks, is outlined for the future. Adaptation of MGWO to multi-objective performance prediction, large-scale global optimization problems, and real-time engineering challenges will be the focus of future research. In bioinformatics, such as DNA and RNA behavior analysis, integrated approach can be used to determine the best feature extraction and filtering approach.

## CONCLUSION

These findings demonstrate that the MGWO outperforms six existing prominent meta-heuristic algorithms. As contrast to other approaches, MGWO showed the best outcomes in 9 benchmark functions, 6 of which are near global optima and 4 of which are the best options. For a number of

Table 2. Running results of functions F1-F23

Function Name	PSOGWO	GWO	MGWO	GWCS	EGWO	AGWO	PSO
	Std	Std	Std	Std	Std	Std	Std
F1	4.15	8.7043e-28	1.5876e-14	4.0829e-27	9.7454e-31	7.192e-43	2.6629e-05
F2	8.61	1.4918e-16	2.2434e-16	1.5232e-17	1.5232e-17	7.9239e-27	0.0050915
F3	9482.1147	7.138e-06	0.0099252	1.4158e-08	0.00079791	3.2187e-09	2251.9731
F4	0.00053149	4.1713e-07	0.00031609	2.2431e-07	0.22196	2.9557e-11	15.7618
F5	26.0403	27.0802	26.175	26.1598	0.7602	0.6465	84.9374
F6	0.0015508	0.50571	9.7619e-05	0.4995	0.5912	0.3273	8.4177e-06
F7	54.9287	0.0040793	0.001101	0.0013597	0.0081689	0.0015895	0.069956
F8	6733.6464	-5599.3474	11691.8935	11686.8541	689.5401	297.2602	-9536.6458
F9	47.773	1.1369e-13	28.7837	5.0971	169.3195	0.21322	56.33
F10	8.4467	1.1102e-13	1.1973e-09	7.5495e-14	0.19456	9.0594e-15	1.0078
F11	0.00019543	0.01374	0	0	0.012937	0.00087209	0.031963
F12	7.8001e-06	0.019186	0.044654	0.037017	3.5249	0.096761	0.65402
F13	26202627.7369	0.26711	1.2175	0.33208	2.4778	1.0243	0.50662
F14	0.99874	10.7632	7.874	2.9821	8.1185	3.2843	1.097
F15	0.00030755	0.020363	0.0013675	0.00031337	0.0037232	0.0032383	0.0033157
F16	-1.0316	-1.0316	-1.0316	-1.0316	-1.0306	-1.0316	-1.0316
F17	26	27.08	26.17	26.98	0.762	0.46	84.37
F18	3	3.0001	3	3	3	8.401	3
F19	-3.8622	-3.8626	-3.8563	-3.8621	-3.8623	-3.8598	-3.8628
F20	-3.322	-3.2024	-3.322	-3.322	-3.2316	-3.1851	-3.2803
F21	-2.6828	-2.6827	-5.1	-10.1481	-6.6414	-6.8244	-5.5507
F22	-9.8568	-10.4021	-5.0877	-10.4009	-7.2449	-7.3594	-7.4128
F23	-10.3517	-10.5351	-5.1752	-10.5353	-7.544	-7.6578	-8.3008

Figure 5a. Three-dimensional images and Simulation results of function F1

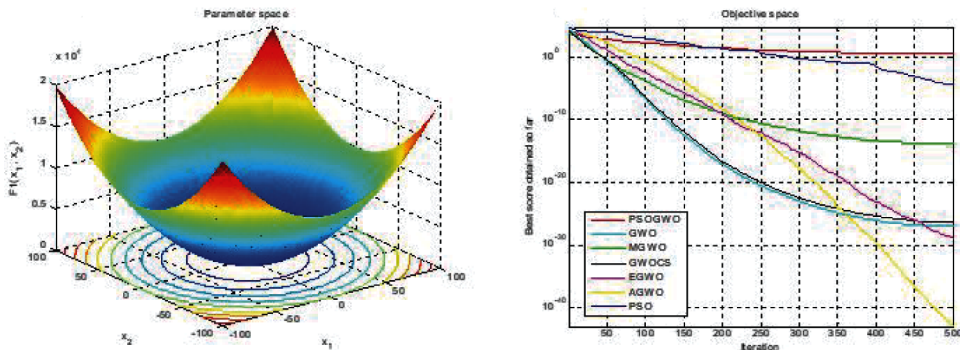




Figure 5b. F2

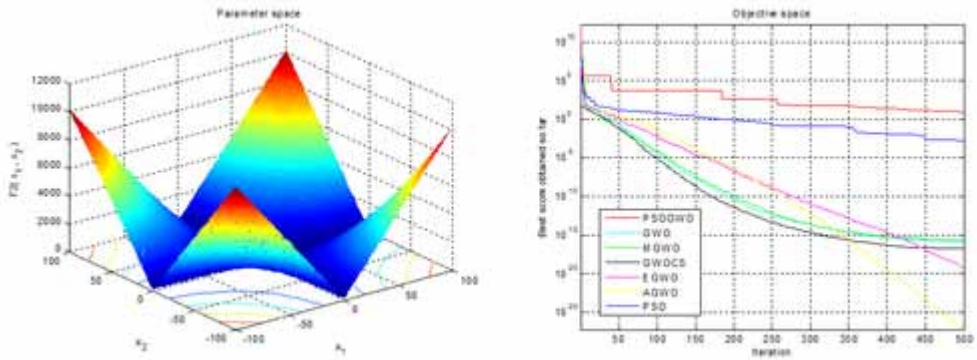


Figure 5c. F3

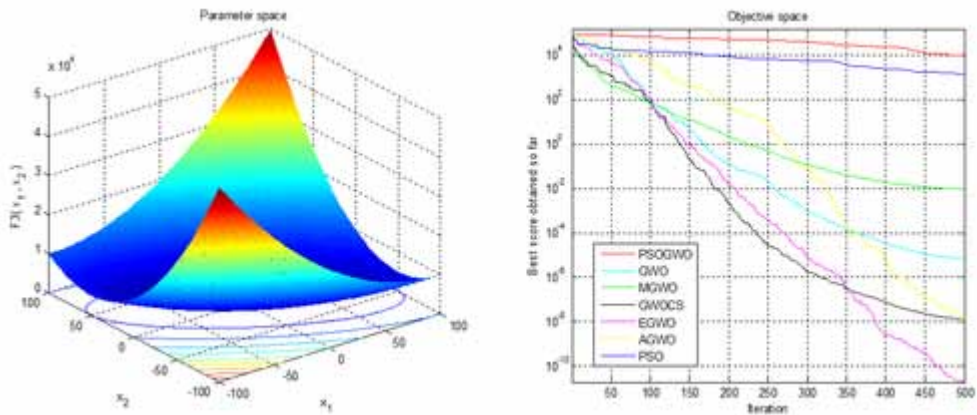


Figure 5d. F4

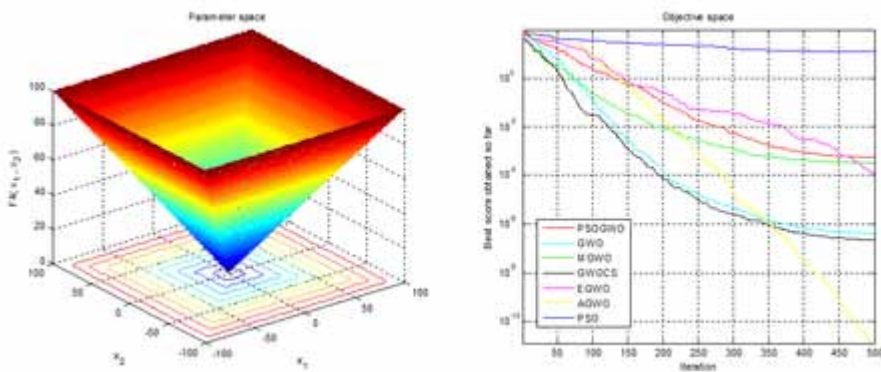


Figure 5e. F5

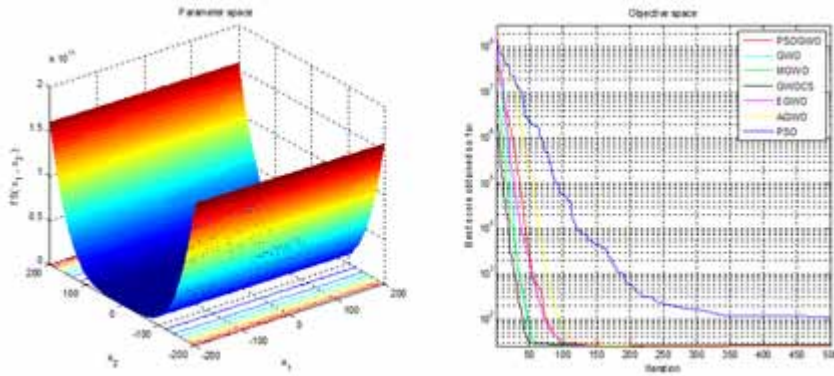


Figure 5f. F6

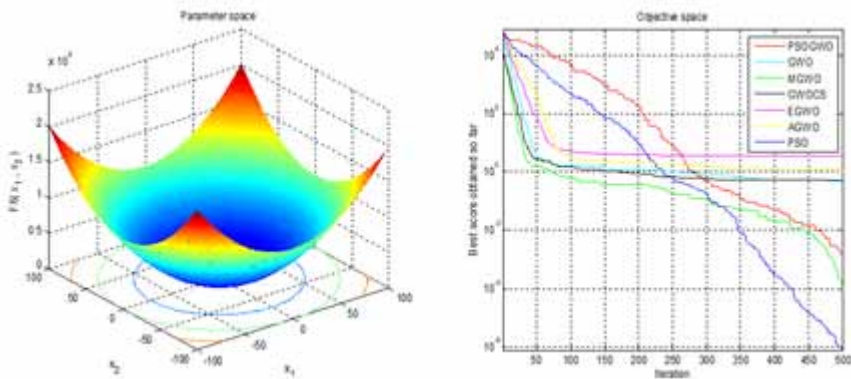


Figure 5g. F7

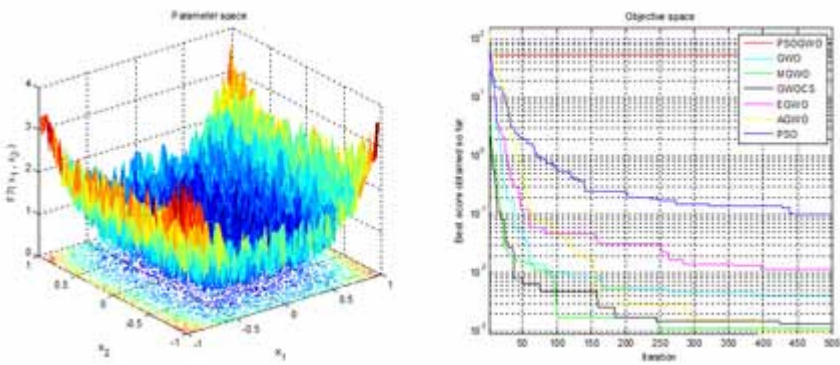


Figure 5h. F8

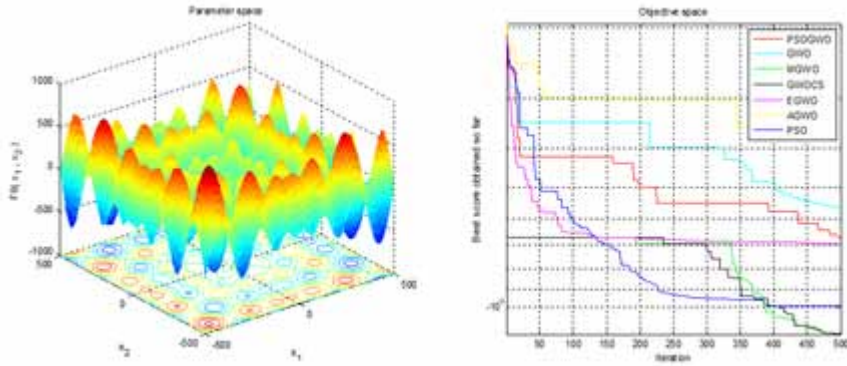


Figure 5i. F9

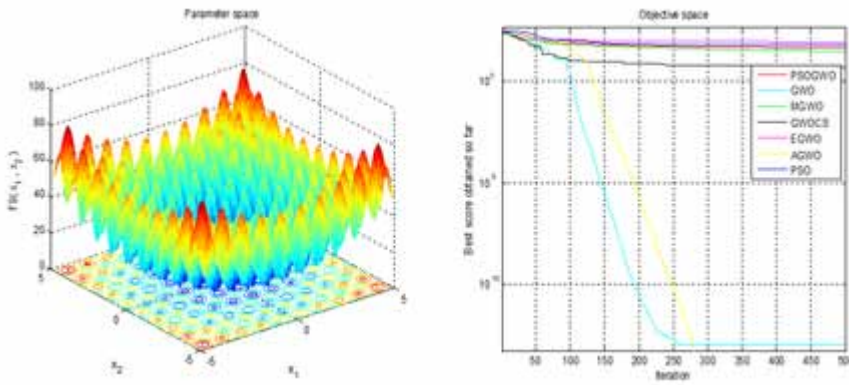


Figure 5j. F10

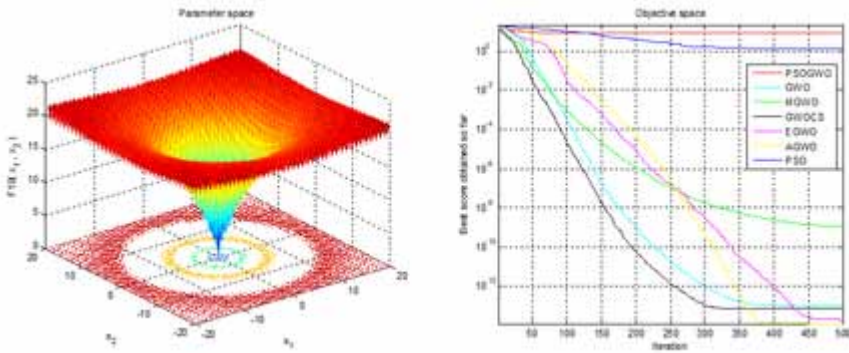


Figure 5k. F11

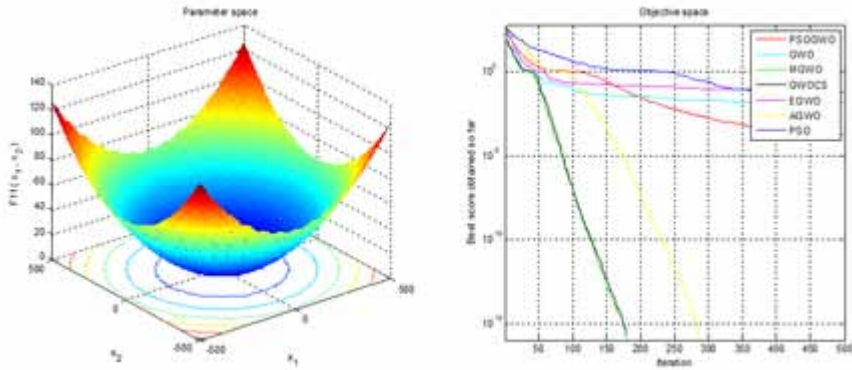


Figure 5l. F12

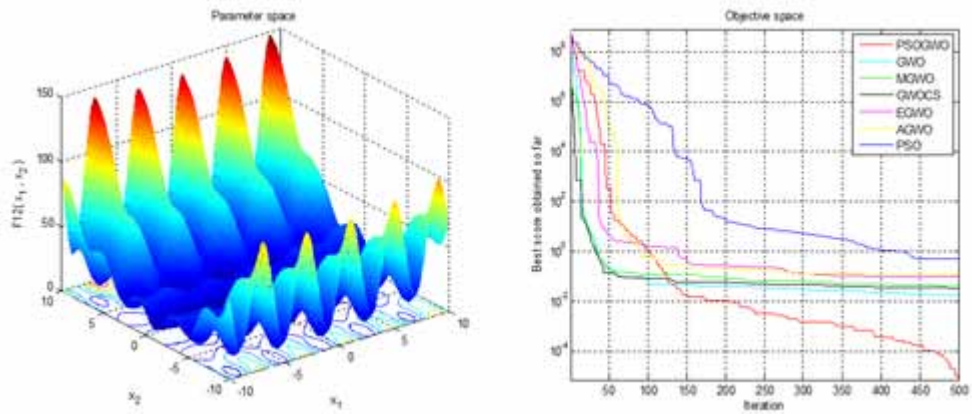


Figure 5m. F13

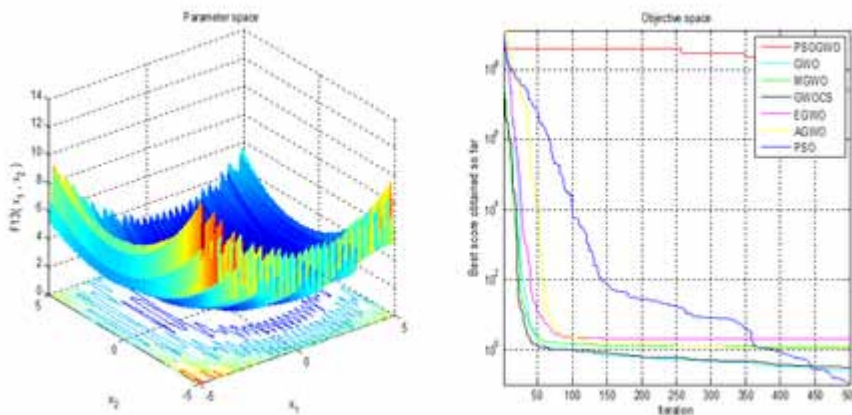


Figure 5n. F14

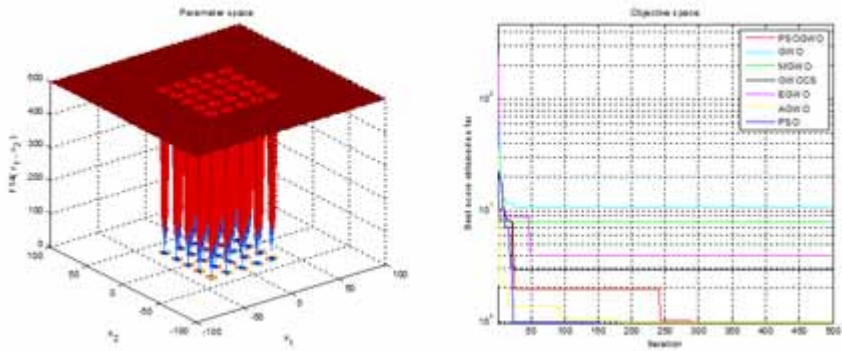


Figure 5o. F15

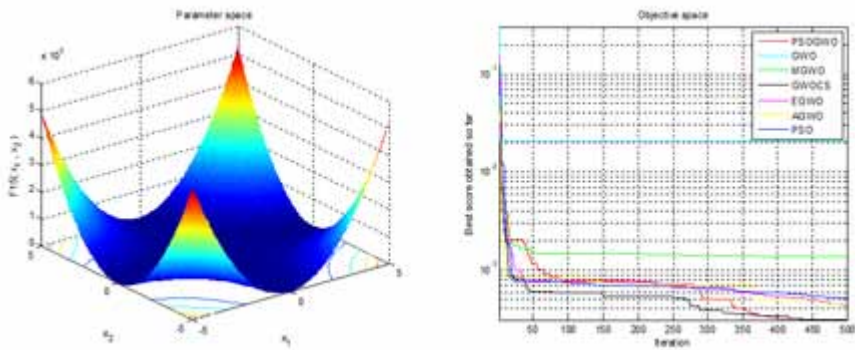


Figure 5p. F18

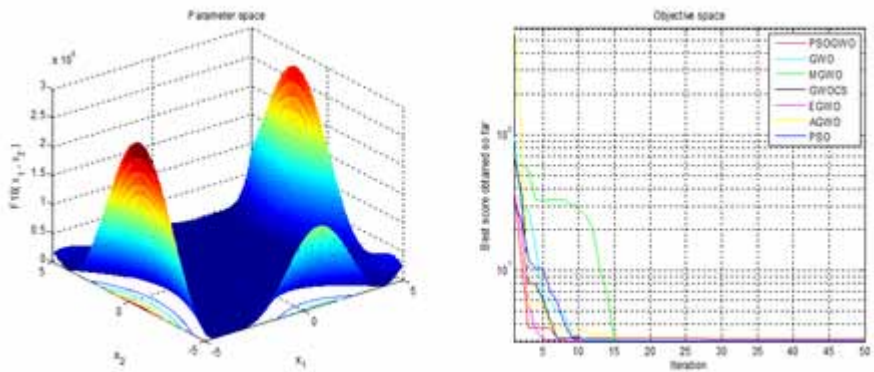


Figure 5q. F19

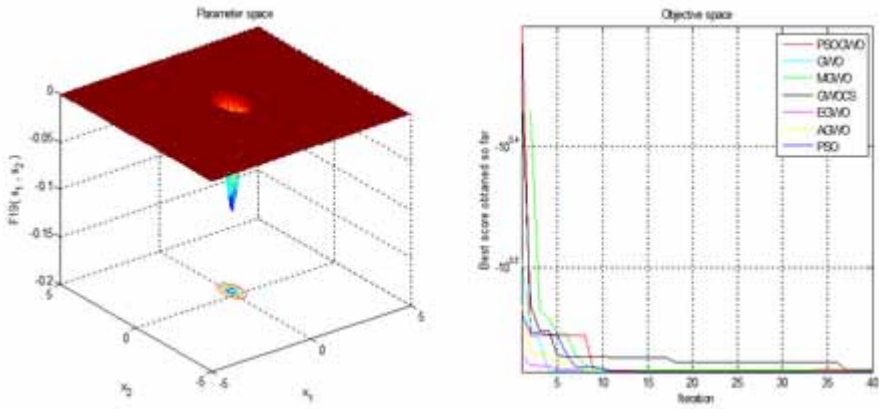


Figure 5r. F20

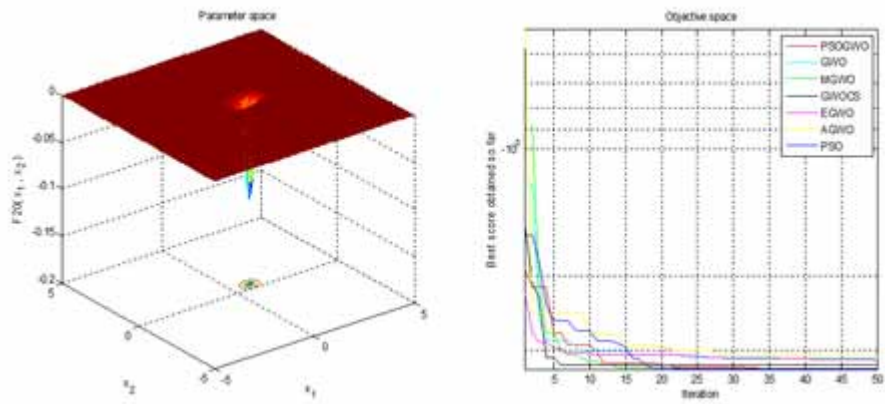


Figure 5s. F21

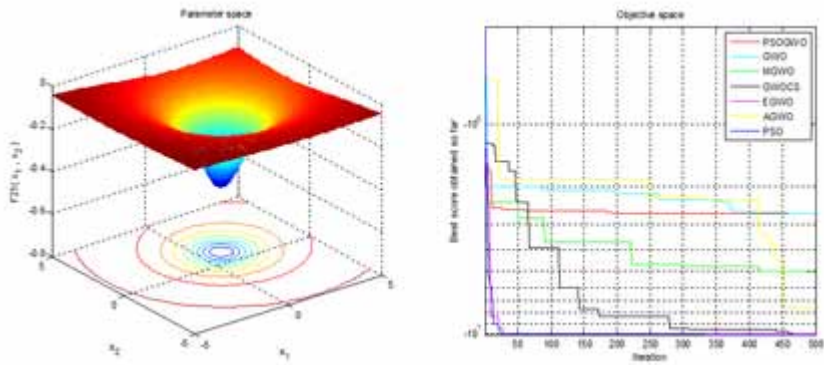


Figure 5t. F22

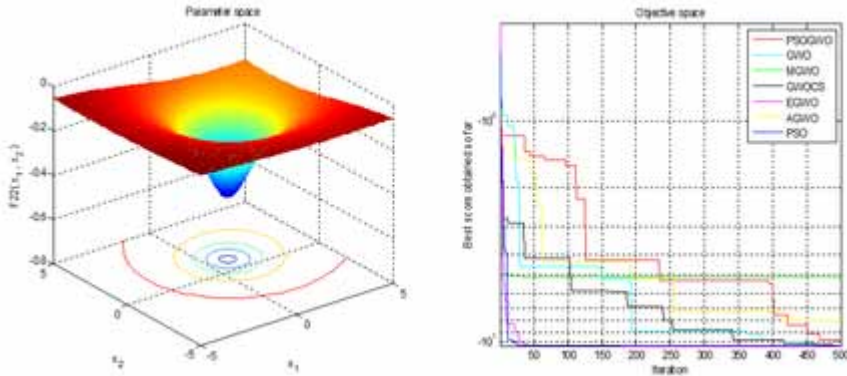
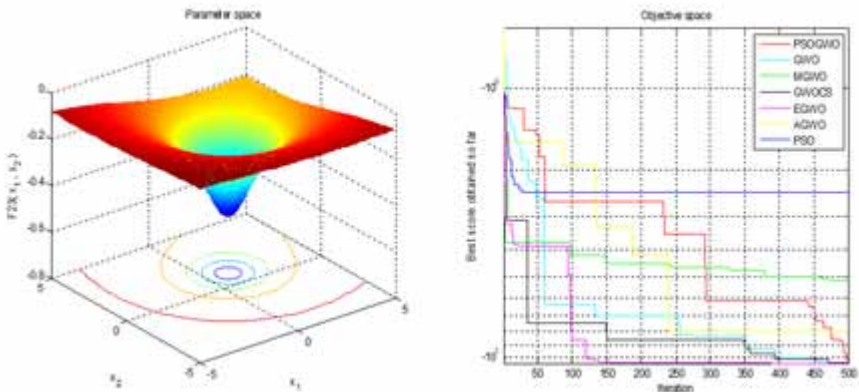


Figure 5u. F23



complicated problems, there is a high likelihood of finding successful solutions. Furthermore, it can easily identify global solutions in a several repetitions. The MGWO approach has an inferior convergence rate to the global solution than other methods, but it has a more consistent behavior and efficiency in many design alternatives. As a consequence, the suggested technique can be implemented to a wide range of applications and challenges to find optimal solutions. MGWO can be utilized in environments where global outcomes are required rapidly and with lower complexity, such as learning-based systems.

Focused on tracking and seeking modes, developed an efficient GWO. On 23 test functions, output of proposed improved strategy is evaluated. In comparison to known heuristics such as GWO, PSO, PSOGWO, EGWO, AGWO, and GWOCS, the analysis indicates that MGWO is sufficient to provide extremely competitive outcomes. The results demonstrate that the MGWO successfully overwhelms the grey wolf optimizer’s lack of local search capacity by striking an equal opportunity for discovery and exploitation. MGWO scores effectively, notably in operations F1, F2, F3, F4, F5, F6, F8, F10, and F12. Similarly, MGWO has decent accuracy to PSOGWO, GWO, GWOCS, EGWO, AGWO, and PSO for F11, F16, F18, F19, and F20. Moreover, F1-F4, F7, F18, and F19 are the closest to the perfect solution, with F11 obtaining the best “0”. The optimization technique in this study has the drawback of not optimizing specific technical challenges. As a result, MGWO improves demographic diversity and reduces the risk of slipping into the local optimum.

## REFERENCES

- Abedi, M., & Gharehchopogh, F. S. (2020). An Improved Opposition Based Learning Firefly Algorithm with Dragonfly Algorithm for Solving Continuous Optimization Problems. *Intelligent Data Analysis*, 24(2), 309–338. doi:10.3233/IDA-194485
- Allahverdi-pour, A., & Gharehchopogh, F. S. (2018). An improved k-nearest neighbour with crow search algorithm for feature selection in text documents classification. *Journal of Advances in Computer Research*, 9(2), 37–48.
- Awad, N. H., Ali, M. Z., & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. *Proc. IEEE Congr. Evol. Comput. (CEC)*, 372-379.
- Fister, I. Jr, Podgorelec, V., & Fister, I. (2021). Improved Nature-Inspired Algorithms for Numeric Association Rule Mining. In P. Vasant, I. Zelinka, & G. W. Weber (Eds.), *Intelligent Computing and Optimization. ICO 2020. Advances in Intelligent Systems and Computing* (Vol. 1324). Springer. doi:10.1007/978-3-030-68154-8\_19
- Ganesan, T., Vasant, P., Sanghvi, P., Thomas, J., & Litvinchev, I. (2020). Random Matrix Generators for Optimizing a Fuzzy Biofuel Supply Chain System. *Journal of Advanced Engineering and Computation*, 4(1).
- Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation*, 48, 1–24. doi:10.1016/j.swevo.2019.03.004
- Gharehchopogh, F. S., Maleki, I., & Talebi, A. (2015). Using Hybrid Model of Artificial Bee Colony and Genetic Algorithms in Software Cost Estimation. *9th International Conference on Application of Information and Communication Technologies (AICT)*, 102-106. doi:10.1109/ICAICT.2015.7338526
- Gharehchopogh, F. S., Shayanfar, H., & Gholizadeh, H. (2019). *A Comprehensive Survey on Symbiotic Organisms Search Algorithms. Artificial Intelligence Review*.
- Guo, M. W., Wang, J. S., Zhu, L. F., Guo, S. S., & Xie, W. (2020). An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems. *IEEE Access*, 8, 69861-69893.
- Hu, P., Pan, J. S., & Chu, S. C. (2020, May). Improved Binary Grey Wolf Optimizer and Its application for feature selection. *Knowledge-Based Systems*, 195(11), 105746. doi:10.1016/j.knosys.2020.105746
- Jamil, M. & Yang, X. S. (2005). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150–194.
- Joshi, H., & Arora, S. (2017). Enhanced Grey Wolf Optimization Algorithm for Global Optimization. *Fundamenta Informaticae*, 153(3), 235–264. doi:10.3233/FI-2017-1539
- Kalita, K., Dey, P., & Haldar, S. (2018). Robust genetically optimized skew laminates. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. doi:10.1177/0954406218756943
- Kalita, K., Dey, P., Haldar, S., & Gao, X.-Z. (2020). Optimizing frequencies of skew composite laminates with metaheuristic algorithms. *Engineering with Computers*, 36(2), 741–761. doi:10.1007/s00366-019-00728-x
- Kalita, K., Ghadai, R. K., & Chakraborty, S. (2021). A comparative study on the metaheuristic-based optimization of skew composite laminates. *Engineering with Computers*. Advance online publication. doi:10.1007/s00366-021-01401-y
- Kalita, K., Mukhopadhyay, T., Dey, P., & Haldar, S. (2020). Genetic programming-assisted multi-scale optimization for multi-objective dynamic performance of laminated composites: The advantage of more elementary-level analyses. *Neural Computing & Applications*, 32(12), 7969–7993. doi:10.1007/s00521-019-04280-z
- Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks*, 1942–1945. doi:10.1109/ICNN.1995.488968
- Khalandi, S., & Gharehchopogh, F. S. (2018). A New Approach for Text Documents Classification with Invasive Weed Optimization and Naive Bayes Classifier. *Journal of Advances in Computer Engineering and Technology, Science and Research Branch*, 4(3), 167–184.



- Kumar, V., Kalita, K., Madhu, S., Ragavendran, U., & Gao, X.-Z. (2021). A Hybrid Genetic Programming–Gray Wolf Optimizer Approach for Process Optimization of Biodiesel Production. *Processes (Basel, Switzerland)*, 9(3), 442. doi:10.3390/pr9030442
- Li, Y., Lin, X., & Liu, J. (2021). An Improved Gray Wolf Optimization Algorithm to Solve Engineering Problems. *Sustainability*, 2021(13), 3208. doi:10.3390/su13063208
- Liang, J. J., Suganthan, P. N., & Deb, K. (2005). Novel composition test functions for numerical global optimization. *Proc. IEEE Swarm Intell. Symp. (SIS)*, 68–75.
- Long, W., Cai, S., Jiao, J., Xu, M., & Wu, T. (2020). A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. *Energy Conversion and Management*, 203, 112243. doi:10.1016/j.enconman.2019.112243
- Majidpour, H., & Gharehchopogh, F. S. (2018). An improved flower pollination algorithm with adaboost algorithm for feature selection in text documents classification. *Journal of Advances in Computer Research*, 9(1), 29–40.
- Mezura-Montes, E., & Coello Coello, C. A. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 1(4), 173–194. doi:10.1016/j.swevo.2011.10.001
- Miandoab, E. E., & Gharehchopogh, F. S. (2017). A novel hybrid algorithm for software cost estimation based on cuckoo optimization and k-nearest neighbors algorithms. *Engineering, Technology & Applied Scientific Research*, 6(3), 1018–1022.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61. doi:10.1016/j.advengsoft.2013.12.007
- Mohammad, H., Shahraki, N., Taghian, S., & Mirjalili, S. (2021). An improved grey wolf optimizer for solving engineering problems. *Expert Systems with Applications*, 166, 113917. doi:10.1016/j.eswa.2020.113917
- Mohammed, H. M., Abdul, Z. K., Rashid, T. A., Alsadoon, A., & Bacanin, N. (2021). A new K-means gray wolf algorithm for engineering problems. *World Journal of Engineering*. doi:10.1108/WJE-10-2020-0527
- Pedersen, M. E. (2010). *Good Parameters for Particle Swarm Optimization*. Hvas Laboratories.
- Prithi, S. & Sumathi, S. (2021). Automata Based Hybrid PSO–GWO Algorithm for Secured Energy Efficient Optimal Routing in Wireless Sensor Network. *Wireless Personal Communications*, 2.
- Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2018). Augmented Grey Wolf Optimizer for Grid connected PMSG-based Wind Energy Conversion Systems. *Applied Soft Computing*, 69, 504–515. doi:10.1016/j.asoc.2018.05.006
- Şenel, F. A., Gökçe, F., Yüksel, A. S., & Yiğit, T. (2019). A novel hybrid PSO–GWO algorithm for optimization problems. *Engineering with Computers*, 35(4), 1359–1373. doi:10.1007/s00366-018-0668-5
- Seyyedabbasi, A., & Kiani, F. (2021). I-GWO and Ex-GWO: Improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. *Engineering with Computers*, 37(1), 509–532. doi:10.1007/s00366-019-00837-7
- Shayanfar, H., & Gharehchopogh, F. S. (2018). Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Applied Soft Computing*, 71, 728–746. doi:10.1016/j.asoc.2018.07.033
- Sumpunsri, S., Thammarat, C., & Puangdownreong, D. (2021). Multiobjective Lévy-Flight Firefly Algorithm for Multiobjective Optimization. In P. Vasant, I. Zelinka, & G. W. Weber (Eds.), *Intelligent Computing and Optimization. ICO 2020. Advances in Intelligent Systems and Computing* (Vol. 1324). Springer. doi:10.1007/978-3-030-68154-8\_15
- Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: An overview. *Soft Computing*, 22(2), 387–408. doi:10.1007/s00500-016-2474-6
- Yue, Z., Zhang, S., & Xiao, W. (2020, April 10). A Novel Hybrid Algorithm Based on Grey Wolf Optimizer and Fireworks Algorithm. *Sensors (Basel)*, 20(7), 2147. doi:10.3390/s20072147

*Vasudha Bahl received her Master's in Technology in Computer Science and Engineering from I.K. Gujral Punjab Technical University, Jalandhar, India and is currently is a PhD Scholar in Department of Computer Science, Banasthali Vidyapith, Rajasthan. Her research activities focused upon Wireless Sensor Networks, Computer Networks and Data Communications.*

*Anoop Kumar Bhola received his Ph.D. from Banasthali Vidyapith, Rajasthan and currently working as Assistant Professor in Computer Science Engineering Department at Banasthali Vidyapith. His main research interests cover Wireless Sensor Networks, Cloud Computing, IoT, and Mobile Computing.*