PAPER • **OPEN ACCESS**

EaGWO: Extended algorithm of Grey Wolf Optimizer

To cite this article: Alok Kumar *et al* 2021 *J. Phys.: Conf. Ser.* **1998** 012028

View the article online for updates and enhancements.

EaGWO: Extended algorithm of Grey Wolf Optimizer

Alok Kumar[1], Lekhraj[1], Avjeet Singh[1] and Anoj Kumar[1]

[1]CSED, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, UP, India

E-mail: alokkumar@mnnit.ac.in

Abstract. The Grey Wolf Optimizer (GWO) and their very recent improved algorithms have some limitations such as stagnation in local optima, slow convergence rate, and so forth. In order to overcome these limitations, we have proposed an algorithm of GWO, named Extended algorithm of GWO (EaGWO), that modifies the hunting equation and encircling equation of the original GWO. The proposed algorithm has been examined on twenty-three very renowned test functions and compared the results with basic GWO and other very recent variants of GWO. This research work also considers the Friedman ANOVA test to verify the effectiveness of the proposed methodology. Consequently, the findings of this work justify quite competitive performance compared to other meta-heuristic algorithms.
Keywords: Optimization Methods, Meta-heuristic Algorithms, Evolutionary Algorithms, Swarm-Based Algorithms, Grey Wolf Optimizer.

1. Introduction

Due to their flexibility, simplicity and local optima avoidance, the meta-heuristic optimization algorithms become quite plausible in the past two decades. These algorithms are applicable to solve various mathematical and real-world problems without any remarkable change in the algorithm's structure that justifies its flexibility. Therefore, the metaheuristic algorithms are treated as a black box to solve the problems.

Further, the meta-heuristic algorithms are straightforward to learn and applied to solve many problems for scientists that refer to the simplicity of algorithms. In addition, these algorithms are typically inspired by evolutionary concepts, animal behaviors, and physics rules that accommodate computer scientists in simulating natural concepts and proposing new meta-heuristic algorithms. Lastly, the meta-heuristic algorithms search extensively looks into the whole search area to prevent the local optima. It offers more capability to avoid local optima to the meta-heuristic algorithms than traditional optimization algorithms.

In addition to the above merits, the meta-heuristic algorithms provide more promising solutions despite incomplete information about the problem, having a limited number of resources, etc. The various optimization algorithms come under meta-heuristic algorithms such as evolutionary algorithms, physics-based algorithms, and swarms-based algorithms. The hierarchical structure of optimization methods has been depicted in Figure 1.

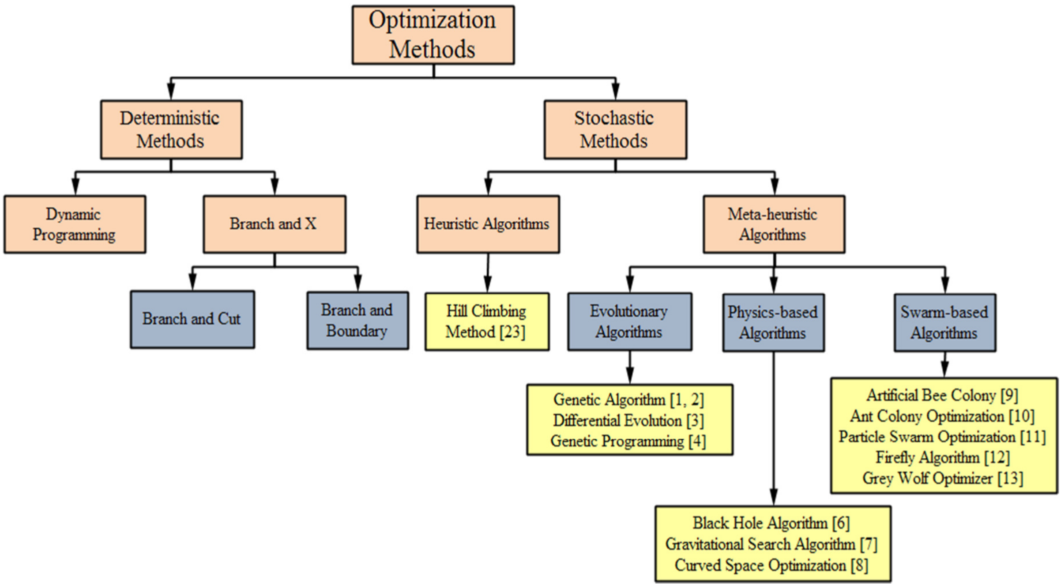

Figure 1. Classification of Optimization Methods.

The various algorithms such as Genetic Algorithm (GA) [1, 2], Differential Evolution (DE) [3] algorithm, Genetic Programming (GP) [4], etc., come under evolutionary algorithms. GA is the most famous and primary algorithm of evolutionary category that Holland proposed. This algorithm simulates the Darwinian evolution concept that refers to genetic behavior and survival of the fitness. The real-world application of genetic algorithm was proposed by Goldberg [5].

Further, physics-based meta-heuristic algorithms are inspired by physics rule. The Black Hole (BH) [6] algorithm, Gravitational Search Algorithm (GSA) [7], Curved Space Optimization (CSO) [8], etc. comes under physics-based algorithms. On the other side, the Artificial Bee Colony (ABC) [9], Ant Colony Optimization (ACO) [10], Particle Swarm Optimization (PSO) [11], Firefly Algorithm (FA) [12], Grey Wolf Optimizer (GWO) [13], etc. are parts of Swarms-based algorithms. Sometimes, the Swarms-based algorithms are also referred Swarm Intelligence (SI) algorithms, implying they are analogous. The PSO algorithm is the most eminent algorithm of this branch. The PSO algorithm was stimulated by the community-based behavior of bird flocking and introduced by Kennedy and Eberhart.

In addition, the GWO is another most famous and very recently developed SI algorithm. The GWO was motivated by the social behavior of the grey wolves and mimicked the leadership hierarchy of them. The GWO and its variants are the key focus of this research article. It may be noted that grey wolf and search agent are analogues throughout in this research article. These algorithms have specific limitations such as slow convergence rate, local optima stagnation, and so forth. Hence, it eventually inspires the researcher to propose another algorithm of GWO to overcome the above limitations.

The primary contributions of this manuscript are summarized as follows:

- Extended social dominant hierarchy is adopted.
- Modified the hunting equation to update the positions of wolves, including omegas.
- Adopt the encircling equation for the suitable equilibrium among exploration and exploitation.
- The proposed algorithm's results justify the effectiveness by comparing with the results of state-of-the-art algorithms such as GWO [13], mGWO [15], MVGWO [18], and WMGWO [20].
- Apply Friedman's ANOVA test to acceptance or rejection of the hypothesis.

The remaining structure of this paper is organized in the following way. A comprehensive discussion about basic GWO and their variants has been presented in Section 2. In Section 3, the proposed algorithm is discussed in the proposed work module in detail. The results and discussion are explained in Section 4, and the research work's conclusion is presented in the last section, i.e. Section 5.

2. Literature Review

In this section, a comprehensive discussion regarding basic grey wolf optimizer and their variants have taken into account. In addition to the above algorithms, the various meta-heuristic optimization techniques have also been discussed.

The Grey Wolf Optimizer (GWO) is a novel and emerging meta-heuristic optimization algorithm that was designed, mathematically formulated, and programmatically implemented by Mirjalili et al. [13] in 2014. The Canis lupus is the scientific name of grey wolves found in North America and Eurasia. The social leadership hierarchy and hunting mechanism of grey wolves are considered the inspiration toward proposing this algorithm. The total populations of grey wolves are divided into four different categories based on their key responsibilities which form the leadership hierarchy with four different levels. The social hierarchy is shown in Figure 2. The group of wolves is called pack. The alpha wolf is the leader of the pack and is called the manager of the pack. It is placed at the apex level of hierarchy. The beta wolf holds at the next lower step. It is called pack discipliner and advisor to alpha. The delta wolf is placed at the second last level of the social hierarchy, and it is reporter to the beta. It is called caretaker to the pack. The remaining wolves are omega wolves, and these are babysitters for the group. In addition to the social hierarchy, three main steps are included in the hunting mechanism. These main steps are searching the prey, harassing the prey until it is tired and stops, and attacking the target at last. The performance has been validated against 29 test functions and four engineering design problems. The optical buffer design, Pressure vessel design, Welded beam design, and Tension/compression spring design are considered engineering design problems. Besides, comparing the obtained results against PSO, GSA, DE, and Fast Evolutionary Programming [14] algorithms have been considered.

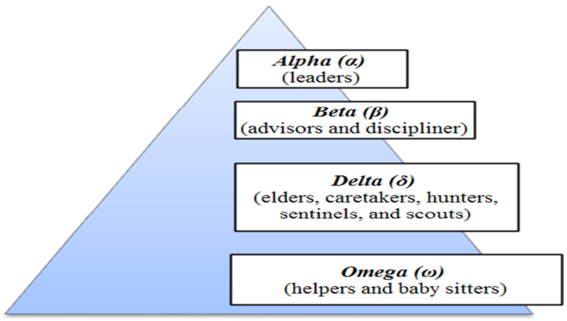

Figure 2. Social hierarchy of grey wolves [13].

To pertinent equilibrium between exploitation and exploration, Mittal et al. [15] proposed a variant of GWO called Modified Grey Wolf Optimizer (mGWO) that employed an exponential decay function instead of a linear function to compute the component of coefficient vector $\vec{a}$ over the course of iterations. In mathematically model of basic GWO, the initial fifty percent components of $\vec{a}$ oblige exploration performance, and the remaining fifty percent devotes to exploitation. In comparison, the proposed approach offers seventy and thirty percent of members dedicated to exploration and exploitation, respectively. The performance of the variant has been benchmarked on 27 test functions and compared the results with PSO, Bat Algorithm (BA) [16], Cuckoo Search (CS) [17], and GWO to validate it. Consequently, the objective results verify the optimal performance of this approach. In addition to the above contribution, the proposed algorithm further employed to cluster head selection in wireless sensor networks and provide the best results compared to comparative well-known meta-heuristic algorithms.

The Modified Variant of GWO (MVGWO) is another algorithm of GWO proposed by Singh et al. [18] to improve the accuracy, performance, and convergence rate. In order to simulate the idea, it considers the total population in five levels of dominant social hierarchy, including omega wolves at the bottom level. The MVGWO employed with the top four levels (delta, gamma, beta, and alpha from bottom to top) of social hierarchy participated in the hunting process. Consequently, this algorithm modifies the encircling behavior and position update equations in the mathematical model. Compared with PSO, GWO, and Mean GWO [19], the objective results verify the algorithm's performance that concludes very competitive outcomes. In addition, the proposed variant has been applied to solve cantilever beam design function and sine data problems that offer very effective solutions against other well-known algorithms.

The Weighted Mean GWO (WMGWO) algorithm has been proposed by Kumar et al. [20] by modifying the hunting mechanism of the original GWO. The proposed algorithm provided the fixed weight to alpha, beta, and delta wolves by 54, 30, and 16 percent, respectively. In addition to the above contribution, it adopts the exponential decay function instead of the linear function from [15]. The proposed algorithm has been benchmarked on twenty-three vary famous classical functions and provides very competitive results compared to GWO, mGWO, and MVGWO algorithms. The algorithm is also benchmarked on classification and function approximation datasets and offers very effective outcomes. In addition, Kumar and Kumar [24] presented a literature review about various optimization techniques and their applications. Furthermore, Kumar et al. [25] enhanced the above literature referred to key focuses on applying the different fields of image processing.

3. Proposed Work

The proposed algorithm (Extended algorithm of Grey Wolf Optimizer (EaGWO)) has been discussed in this section.

3.1. *The EaGWO Algorithm*

To obtain the objective, the evaluation begins with a random population of grey wolves (search agents) of various levels of social hierarchy. The proposed work adopts the dominant social hierarchy from [18]. Hence, the total population is currently divided into five groups. The alpha (α) wolf is the apex wolf of social hierarchy and is regarded as the pack leader, the decision-maker for the pack, and the most denominated wolf in the group. In a mathematical model, it is the most promising solution to the problem. The beta wolf (β) exists at the next lower level of the hierarchy that is an advisor to the alpha and is considered as the second-best solution to the problem. The gamma (γ) and delta (δ) wolves are placed at the third and fourth levels of the social hierarchy, respectively, which helps hunting the prey and other related activities for the pack. The remaining wolves are omega (ω) that have been involved at the bottom level and treated as the lowest-ranked for the pack.

In order to the mathematical model, alpha, beta, gamma, and delta wolves are considered four random best solutions from initial random solutions. The proposed algorithm is implemented through three groups to modify the hunting behavior of grey wolves. These groups are formed from alpha, beta, gamma, and delta wolves. These three groups indicate three different probable positions of the prey, and the final position of the prey is computed using equation (5). It may be noted that grey wolf and search agent are analogues throughout in this research article.

Therefore, the alpha, beta, gamma, and delta wolves are involved in forming the first group, which indicate $\overrightarrow{X_1}$, $\overrightarrow{X_2}$, $\overrightarrow{X_3}$, and $\overrightarrow{X_4}$ probable positions of prey, respectively and the average value of indicated positions computes the first expected position of the prey ($\vec{X}_\sigma$) that is calculated using equation (1).

$$\vec{X}_\sigma = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}+\overrightarrow{X_4}}{4} \tag{1}$$

Here,

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.(\overrightarrow{D_\alpha})$$
$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.(\overrightarrow{D_\beta})$$
$$\overrightarrow{X_3} = \overrightarrow{X_\gamma} - \overrightarrow{A_3}.(\overrightarrow{D_\gamma})$$
$$\overrightarrow{X_4} = \overrightarrow{X_\delta} - \overrightarrow{A_4}.(\overrightarrow{D_\delta})$$

The $\vec{D}$ is the distance vector that indicates distance between the prey's current position and the corresponding wolf, for instance $\overrightarrow{D_\alpha}$ shows the distance between current position of the prey and the alpha wolf. The $\overrightarrow{D_\alpha}$ is computed as follows:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}\right|$$

Here, $\vec{X}$ is the current position of the prey, whereas $\overrightarrow{C_1}$ is called as coefficient vector which is calculated as follows:

$$\overrightarrow{C_1} = 2.\overrightarrow{r_1}$$

The $\vec{A}$ is also known as coefficient vector of corresponding wolf, such as $\overrightarrow{A_1}$ is the coefficient vectors of alpha wolf that is computed as follows:

$$\overrightarrow{A_1} = 2.\vec{a}.\vec{r}_1 - \vec{a}$$

Similarly, $\overrightarrow{A_2}$, $\overrightarrow{A_3}$, and $\overrightarrow{A_4}$ are the coefficient vectors regard with beta, gamma, and delta wolves, respectively and calculated in the similar way. Whereas, $\vec{a}$ decreased exponentially [15] over the course of iteration from 2 to 0 (computed using equation (2)) concludes that the initial 70 percent components decreasing quite slowly, implying provided more diversity (extensive exploration) to the proposed algorithm. On the other hand, the remaining 30 percent components decrease quickly that implied a fast convergence rate (quick exploitation) to the proposed algorithm.

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

t: Current iteration
T: Maximum number of iterations

In addition, the only alpha, beta, and gamma wolves are involved in forming the second group this time, which indicate $\overrightarrow{X_5}$, $\overrightarrow{X_6}$, and $\overrightarrow{X_7}$ probable positions of prey, respectively and the average value of indicated positions computes the second expected position of the prey ($\vec{X}_{\sigma\_1}$) that is calculated using equation (3).

$$\vec{X}_{\sigma\_1} = \frac{\overrightarrow{X_5}+\overrightarrow{X_6}+\overrightarrow{X_7}}{3} \tag{3}$$

Here,

$$\overrightarrow{X_5} = \overrightarrow{X_\alpha} - \overrightarrow{A_5}.(\overrightarrow{D_\alpha})$$
$$\overrightarrow{X_6} = \overrightarrow{X_\beta} - \overrightarrow{A_6}.(\overrightarrow{D_\beta})$$

$$\overrightarrow{X_7} = \overrightarrow{X_\gamma} - \overrightarrow{A_7}.(\overrightarrow{D_\gamma})$$

Successively, only beta, gamma, and delta wolves are involved in forming the third group in last time, which indicate $\overrightarrow{X_8}$, $\overrightarrow{X_9}$, and $\overrightarrow{X_{10}}$ positions of prey, respectively and the average value of indicated positions computes the third expected location of the prey ($\vec{X}_{\sigma\_2}$) that is calculated using equation (4).

$$\vec{X}_{\sigma\_2} = \frac{\overrightarrow{X_8}+\overrightarrow{X_9}+\overrightarrow{X_{10}}}{3} \tag{4}$$

Here,

$$\overrightarrow{X_8} = \overrightarrow{X_\beta} - \overrightarrow{A_8}.(\overrightarrow{D_\beta})$$
$$\overrightarrow{X_9} = \overrightarrow{X_\gamma} - \overrightarrow{A_9}.(\overrightarrow{D_\gamma})$$
$$\overrightarrow{X_{10}} = \overrightarrow{X_\delta} - \overrightarrow{A_{10}}.(\overrightarrow{D_\delta})$$

Finally, the first, second, and third expected positions eventually calculate the prey's final position computed using equation (5). All wolves update its position according to prey's position.

$$\vec{X}(t+1) = \vec{X}_\sigma + \vec{r}.(\vec{X}_{\sigma\_1} - \vec{X}_{\sigma\_2}) \tag{5}$$

Where '$\vec{r}$' is random vector contains the random value between 0 and 1. The above steps are repeated till the maximum number of iterations. Consequently, alpha's position at the end of last iteration is considered as the final solution of the algorithm to the problem. Figure 3 depicted the pseudo code of the proposed algorithm.

Initialize the population of grey wolves (Search Agents) $\overrightarrow{X_k}$ (k = 1, 2, 3... n)
Initialize controlling parameters $\vec{a}$, $\vec{A}$, and $\vec{C}$
Compute the fitness of each search agent
$\overrightarrow{X_\alpha}$ = the most fittest solution from all search agents
$\overrightarrow{X_\beta}$ = the second best solution from all search agents
$\overrightarrow{X_\gamma}$= the third best solution from all search agents
$\overrightarrow{X_\delta}$ = the fourth best solution from all search agents
while (t < Maximum number of iterations) **do**
  for each search agent ($\overrightarrow{X_k}$) **do**
    Update the position of current $\overrightarrow{X_k}$ using equation **(5)**
  end for
  Update the value of controlling parameters $\vec{a}$ (using equation **2**), $\vec{A}$, and $\vec{C}$
  Compute the fitness of all search agents
  Update the value of $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, $\overrightarrow{X_\gamma}$ and $\overrightarrow{X_\delta}$ **(for each group)**
  t = t + 1
end while
return $\overrightarrow{X_\alpha}$

Figure 3. Pseudo code of the EaGWO algorithm.

4. Results and Discussion

A detailed description of benchmarked test functions and the proposed algorithm's obtained performance has been discussed in this section. The twenty-three very famous benchmarked functions [21] are involved in validating the performance with a range of dimensionalities and a variety of optimal solutions. These test functions are categorized into three categories: fixed-dimension multimodal (ten functions), multimodal (six functions), and unimodal (seven functions)

functions. It may be noted, the first seven unimodal functions (F1-F7) are included, which have single optima that validate the exploitation performance of the proposed algorithm. In contrast, to validate the exploration performance, six multimodal functions (F8-F13) are utilized, which have a massive number of local optima. On the other hand, ten fixed-dimension multimodal functions (F14-F23) validate the exploration and avoidance of local optima. Interestingly, the proposed research work considers various functions to validate the strength of the proposed algorithm. The experimental environment is tabulated in the Table 1.

Table 1. Experimental environment.

Parameter	Hardware and Software Configuration
Implementation Tool	MATLAB R2017a
Operating System	64-bit Operating System
Processor	Intel(R) Core(TM) i7-4770 CPU@ 3.40GHz
RAM	12.0 GB

In this implementation, the initial populations (search agents or grey wolves) are 30 which are randomly created and the above steps have been repeated iteratively 500 times to compute the optimal value in one independent run of the proposed algorithm in practice. The algorithm repeats 30 times for each benchmark function. Hence, the average (Mean) of 30 independent runs concludes the final output value of the corresponding benchmark function. The other statistics variables (Best, Worst, and Std) are also utilized to validate the outcomes. The Best variable shows the minimum value throughout the 30 independent runs, whereas the Worst indicates the maximum value. In addition, the Std represents the standard deviation, and that is computed through 30 separate runs. The benchmark functions are minimization functions. Hence, the minimum value of all statistics variables is considered the optimum value of the proposed algorithm concerning the corresponding benchmark function. The results are compared with the original GWO, mGWO, MVGWO, and WMGWO to verify the effectiveness of the proposed algorithm. We deliberately compared it only with basic GWO and their variants because the GWO has already been compared with PSO, GSA, DE, and fast evolutionary programming and offers very competitive results. The mathematical results are tabulated in Table (2), (3), and (4), and bold values show the most optimal outcomes.

From all test beds, it should be noticed that there are only seven unimodal functions (F1-F7) that contain single optima. These unimodal functions assure the exploitation performance of the suggested algorithm against GWO's other well-known variants and original GWO. The results of unimodal functions are shown in Table 2. The proposed algorithm outperforms on F5 and F6 functions against all comparative algorithms and gives competitive results for additional functions. Consequently, the proposed approach provides significantly better exploitation than other existing algorithms.

Contrary to the unimodal functions, multimodal functions contain an enormous number of local optima utilized to evaluate the exploration performance. In this study, there are six multimodal functions (F8-F13), for which results are listed in Table 3. According to this table, the suggested technique performs superior to comparative algorithms on F11, F12, and F13 benchmarked and achieved highly competitive outcomes over other multimodal functions. Therefore, the proposed algorithm delivers relatively excellent exploration in comparison to other comparative methods.

Finally, the ultimate ten mathematical benchmark functions are fixed-dimension multimodal functions (F14-F23), investigating local optima avoidance and extensive exploration. The computational results of these functions are listed in Table 4. The experimental study demonstrates that the EaGWO algorithm is vastly superior to all comparable algorithms for F14, F18, and F23 multimodal functions and delivers highly competitive results against the remaining functions.

Consequently, from the comparative analysis of the experimental approaches, the proposed algorithm significantly avoids the local optima and obtained reasonable exploration.

Table 2. Results of unimodal functions.

Functions	Criteria	GWO	mGWO	MVGWO	WMGWO	EaGWO
F1	Best	1.86955E-29	2.47567E-38	9.0964E-23	**3.65451E-40**	5.67861E-31
	Worst	4.71526E-27	5.04277E-35	2.57737E-20	**6.39631E-37**	5.23137E-28
	Mean	1.21491E-27	3.56624E-36	7.86559E-21	**7.02897E-38**	6.31315E-29
	Std	1.5077E-27	9.31078E-36	7.59709E-21	**1.31945E-37**	1.20624E-28
F2	Best	1.1563E-17	5.63425E-23	1.70886E-13	**8.98768E-24**	1.63081E-18
	Worst	3.51893E-16	3.80217E-21	2.33917E-12	**5.35169E-22**	8.55475E-17
	Mean	1.09926E-16	1.00731E-21	1.11925E-12	**1.03797E-22**	1.7488E-17
	Std	8.80098E-17	8.26037E-22	6.02624E-13	**1.18008E-22**	1.69863E-17
F3	Best	3.0341E-08	1.65536E-11	1.21667E-07	**1.23199E-11**	5.21289E-06
	Worst	4.87427E-05	1.19346E-05	0.007249688	**1.29726E-06**	0.003085718
	Mean	6.29046E-06	4.65057E-07	0.000644064	**9.79761E-08**	0.00049894
	Std	1.22937E-05	2.17128E-06	0.001397923	**2.77361E-07**	0.000830755
F4	Best	2.08233E-07	8.60812E-11	7.33913E-06	**3.20701E-11**	3.08081E-08
	Worst	3.74257E-06	**3.14336E-09**	0.000261329	1.9422E-08	1.69252E-06
	Mean	7.60116E-07	**1.03809E-09**	3.69567E-05	2.17758E-09	4.1377E-07
	Std	7.46484E-07	**9.24871E-10**	4.7946E-05	3.81168E-09	4.40451E-07
F5	Best	26.23472593	25.50392158	26.00537781	25.7963667	**25.37336031**
	Worst	28.54785126	27.18916864	28.85103417	28.74838882	**27.17470325**
	Mean	27.24082871	26.54974036	27.45641445	26.83712342	**26.41503256**
	Std	0.541230932	0.481869522	0.886454746	0.62899106	**0.456054025**
F6	Best	**8.81922E-05**	0.000209862	0.462555614	0.000337482	0.000191879
	Worst	1.504360811	1.374088192	2.500416269	1.504592156	**0.966875562**
	Mean	0.836171787	0.592375132	1.37762182	0.737688404	**0.379991954**
	Std	0.419552688	0.373744731	0.479311786	0.350429673	**0.29380457**
F7	Best	0.00036451	0.000255995	0.00068187	**0.000208123**	0.000409668
	Worst	0.010308008	**0.003761961**	0.00578016	0.005615419	0.005215894
	Mean	0.002162786	**0.001473201**	0.002637996	0.001507031	0.001833639
	Std	0.001918486	**0.000847617**	0.001137545	0.001140975	0.001263831

Table 3. Results of multimodal functions.

Functions	Criteria	GWO	mGWO	MVGWO	WMGWO	EaGWO
F8	Best	-7007.067326	-7324.40433	-7365.397337	-7133.682276	**-7803.430059**
	Worst	-3451.878595	-3340.825249	**-3971.721559**	-3452.826614	-3377.682692
	Mean	-5779.214936	-5730.844032	-5840.220604	-5603.842752	**-5940.057013**
	Std	863.2310254	1030.629966	**792.3632517**	947.4670136	1422.918648
F9	Best	5.68434E-14	**0**	1.76215E-12	**0**	**0**
	Worst	15.46592098	**2.224124044**	15.84489418	7.945244387	4.092774197
	Mean	1.816528235	**0.074137469**	7.395227793	0.44404023	0.332263608
	Std	3.664188042	**0.406067636**	3.909905779	1.723196002	0.952318997
F10	Best	6.4837E-14	1.5099E-14	3.69216E-12	**1.15463E-14**	3.9968E-14
	Worst	1.50102E-13	**2.93099E-14**	4.62288E-11	**2.93099E-14**	6.4837E-14
	Mean	1.03088E-13	2.18492E-14	1.52304E-11	**1.60464E-14**	4.67182E-14
	Std	1.95189E-14	3.40892E-15	1.05978E-11	**3.3553E-15**	8.31093E-15
F11	Best	**0**	**0**	**0**	**0**	**0**
	Worst	0.029200934	**0.015904907**	0.030505877	0.016767589	0.019830261
	Mean	0.005790525	0.001482942	0.005565115	0.001261958	**0.000661009**
	Std	0.009743901	0.004531504	0.010554725	0.003990152	**0.003620494**
F12	Best	0.019524391	0.01326536	0.03129131	0.020261853	**0.006730561**
	Worst	0.097500844	0.082938209	0.244762389	0.107450774	**0.063864821**
	Mean	0.046806318	0.044951636	0.081413055	0.046465817	**0.027962762**
	Std	0.022134319	0.018554439	0.054746096	0.0184263	**0.013369255**
F13	Best	**0.000149445**	0.000603853	0.580348302	0.414048257	0.099982513

	Worst	1.038006476	0.818851025	1.416093938	1.062261347	**0.58505613**
	Mean	0.59374835	0.444310005	1.03498469	0.632473804	**0.256849957**
	Std	0.260627449	0.171736169	0.266188319	0.157637637	**0.131901913**

Table 4. Results of fixed-dimension multimodal functions.

Functions	Criteria	GWO	mGWO	MVGWO	WMGWO	EaGWO
F14	Best	**0.998003838**	**0.998003838**	**0.998003838**	**0.998003838**	**0.998003838**
	Worst	12.67050581	12.67050581	12.67050581	12.67050581	**10.76318067**
	Mean	5.521999671	4.970329775	4.913586384	3.350477921	**2.305205074**
	Std	4.97107733	4.811325311	4.332521896	3.672002965	**2.962385301**
F15	Best	**0.000307495**	0.000307509	0.000307672	0.000307498	0.000308529
	Worst	**0.020363344**	0.020363352	0.020363357	0.056542965	0.020363368
	Mean	0.0017574	0.00505625	0.003742333	0.004964009	**0.001185079**
	Std	0.005062792	0.008589344	0.007562054	**0.011922326**	0.003626359
F16	Best	-1.031628451	-1.031628452	**-1.031628453**	**-1.031628453**	-1.031628452
	Worst	**-1.031628377**	-1.031628216	-1.031628372	-1.031628098	-1.031628207
	Mean	-1.031628434	-1.031628382	**-1.031628435**	-1.031628344	-1.03162837
	Std	1.95414E-08	5.77609E-08	2.03588E-08	**1.05912E-07**	7.74589E-08
F17	Best	0.397887374	0.397887448	**0.397887359**	0.397887675	0.397887603
	Worst	0.408765443	0.399999963	0.399381015	**0.397961634**	0.39826969
	Mean	0.398251605	0.397967692	0.397938963	**0.3978981**	0.397914828
	Std	0.001985748	0.000384124	0.000272375	**1.65123E-05**	7.07351E-05
F18	Best	3.00000008	3.00000017	3.000000455	3.000000017	**3.000000004**
	Worst	84.00001954	3.00020306	3.000298014	3.000094621	**3.000035893**
	Mean	5.700058402	3.000026888	3.000070194	3.000017236	**3.000008849**
	Std	14.78850171	4.1112E-05	8.60202E-05	2.29469E-05	**1.08025E-05**
F19	Best	-3.862780166	-3.862776433	**-3.862782056**	-3.862768948	-3.862768644
	Worst	-3.854894755	-3.855829157	**-3.856653839**	-3.854894638	-3.854893719
	Mean	-3.861094751	-3.861799343	**-3.862112723**	-3.861068162	-3.861251461
	Std	0.002698021	0.001785986	**0.001630442**	0.002821052	0.002696434
F20	Best	**-3.321993943**	-3.321991373	-3.321993629	-3.321982237	-3.321975576
	Worst	-3.03572696	-3.086683302	**-3.197572676**	-3.016684639	-3.133093166
	Mean	-3.281405865	-3.244678409	-3.281901632	-3.224314734	**-3.285430714**
	Std	0.078966025	0.079250315	**0.057665614**	0.091073266	0.063154158
F21	Best	-10.15272915	-10.15149601	**-10.15307601**	-10.15125096	-10.14978048
	Worst	-2.630377954	-2.68128583	-2.630288681	-2.629751345	**-5.054800861**
	Mean	-8.65238314	**-9.896895483**	-9.068632616	-8.383141593	-9.633628289
	Std	2.57762199	**1.362819231**	2.511493824	2.810193436	1.552312306
F22	Best	-10.4024973	-10.40113809	**-10.40279949**	-10.40230787	-10.40132551
	Worst	-5.08765936	-5.087655934	-5.087650497	**-10.38485698**	-10.38026789
	Mean	-10.22417062	-10.21859117	-10.22437885	**-10.39456127**	-10.39282998
	Std	0.970132445	0.969086409	0.97017339	0.005428945	**0.00521214**
F23	Best	-10.53632168	-10.53600482	-10.53597644	-10.53496829	**-10.53789908**
	Worst	-2.421642548	-2.421550213	-2.421731517	-2.421295029	**-5.15961031**
	Mean	-9.813553362	-10.07977871	-10.26464148	-9.808373231	**-10.34773367**
	Std	2.238139156	1.750305312	1.481289373	2.236482789	**0.979896829**

4.1. *Friedman's ANOVA Test*

The null hypothesis $[H_0]$ states that all algorithms perform equivalently regarding every problem aspect. In contrast, the alternative hypothesis $[H_1]$ claims that all algorithms are not equivalent. Therefore, the objective has to reject the null hypothesis and accept the alternative hypothesis. It is validated through the comparative study of performance regarding existing algorithms and the proposed algorithm. Hence, the Friedman test has been utilized for this objective in this research work. The Friedman test was performed as a non-parametric test equivalent to the repeatedmeasures ANOVA [22]. The mathematical formula is formulated using equation (6) to calculate the Friedman statistic.

$$F_V = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1) \quad (6)$$

In equation (6), F_V indicates Friedman statistic (value of Friedman test), k and n are indicating number of algorithms and number of benchmark functions, respectively. The R_j holds the rank sum of j[th] algorithm.

Friedman's ANOVA Table

Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	57.217	4	14.3043	22.89	0.0001
Error	172.783	88	1.9634		
Total	230	114			

Test for column effects after row effects are removed

Figure 4. Friedman's ANOVA test result.

In the current implementation of the EaGWO algorithm, the value of k is five, and the value of n is 23. The obtained value of Friedman chi-square statistics is 22.89 and the P-value for the Friedman's chi-square statistics is 0.0001, as shown in Figure 4. Hence, P-value for the Friedman's chi-square statistics result is more diminutive than Friedman's chi-square statistics that reject the null hypothesis, implying all algorithms are not equivalent. In order to concerning individual benchmarked function, the average ranks 3.83, 2.61, 3.74, 2.87, and 1.96 are obtained by GWO, mGWO, MVGWO, WMGWO, and EaGWO algorithms, respectively. The average rank of the proposed algorithm is much better than other variant's average rank hence also rejects the null hypothesis. The overall rank of the proposed algorithm is one, which means it outperforms the other comparative algorithms. The ranks of all algorithms based on mean value of results of each benchmark function are tabulated in Table 5.

Table 5. Rank of algorithms based on mean value of results.

Benchmark Functions	GWO	mGWO	MVGWO	WMGWO	EaGWO
F1	4	2	5	1	3
F2	4	2	5	1	3
F3	3	2	5	1	4
F4	4	1	5	2	3
F5	4	2	5	3	1
F6	4	2	5	3	1
F7	4	1	5	2	3
F8	3	4	2	5	1
F9	4	1	5	3	2
F10	4	2	5	1	3
F11	5	3	4	2	1
F12	4	2	5	3	1
F13	3	2	5	4	1
F14	5	4	3	2	1
F15	2	5	3	4	1
F16	2	3	1	5	4
F17	5	4	3	1	2
F18	5	3	4	2	1
F19	4	2	1	5	3
F20	3	4	2	5	1

F21	4	1	3	5	2
F22	4	5	3	1	2
F23	4	3	2	5	1
Rank's Sum	88	60	86	66	45
Average Rank	3.83	2.61	3.74	2.87	1.96
Overall Rank	5	2	4	3	1

5. Conclusion

This research work recommended a novel variant of GWO entitled, Extended algorithm of GWO (EaGWO) in order to magnify the convergence rate and help prevent the local optima stagnation. The proposed algorithm reforms the hunting equation and adaptations of the encircling equation from the mGWO to attain the objective. The twenty-three benchmark functions have been utilized to analyzing the proposed algorithm's performance. The proposed algorithm outperforms the majority of the benchmark functions and provides quite respectable results for the remaining functions, particularly compared to the traditional GWO and other state-of-the-art their algorithms. Moreover, Friedman's ANOVA test has been applied that rejects the null hypothesis and accepts the alternative hypothesis, which confirmed the efficacy of the proposed algorithm. For future work, the proposed algorithm will be addressed to tackle the real-life problem, which is currently being under development.

References

[1] Holland J H 1992 Genetic algorithms *Scientific american* **267**(1) pp. 66-73.
[2] Krishnakumar K and Goldberg D E 1992 Control system optimization using genetic algorithms *Journal of Guidance, Control, and Dynamics* **15**(3), pp. 735-740.
[3] Storn R and Price K 1997 Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces *Journal of global optimization* **11**(4) pp. 341-359.
[4] Koza J R and Koza J R 1992 Genetic programming: on the programming of computers by means of natural selection (Vol. 1) *MIT press*.
[5] Goldberg D 1994 Genetic algorithms in optimization, search and machine learning *Reading*.
[6] Hatamlou A 2013 Black hole: A new heuristic optimization approach for data clustering *Information sciences* **222** pp. 175-184.
[7] Rashedi E, Nezamabadi-Pour H and Saryazdi S 2009 GSA: a gravitational search algorithm Information sciences **179**(13) pp. 2232-2248.
[8] Moghaddam F F, Moghaddam R F, and Cheriet M 2012 Curved space optimization: a random search based on general relativity theory *arXiv preprint arXiv:1208.2214*.
[9] Basturk B and Karaboga D 2006 An artificial bee colony (ABC) algorithm for numeric function optimization *In: IEEE swarm intelligence symposium* pp. 12–4.
[10] Dorigo M, Birattari M and Stutzle T 2006 Ant colony optimization *IEEE computational intelligence magazine* **1**(4) pp. 28-39.
[11] Kennedy J and Eberhart R 1995 November. Particle swarm optimization Proceedings of ICNN'95-international conference on neural networks (Vol. 4 pp. 1942-1948). IEEE.
[12] Yang X S 2010 Firefly algorithm, stochastic test functions and design optimisation *International journal of bio-inspired computation* **2**(2) pp. 78-84.
[13] Mirjalili S, Mirjalili S M and Lewis A 2014 Grey wolf optimizer Advances in engineering software 69 pp. 46-61.
[14] Yao X and Liu Y 1996 Fast Evolutionary Programming *Evolutionary programming* **3** pp. 451-460.
[15] Mittal N, Singh U, and Sohi B S 2016 Modified grey wolf optimizer for global engineering optimization *Applied Computational Intelligence and Soft Computing*.

[16] Yang X S 2010 A new metaheuristic bat-inspired algorithm *Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Springer, Berlin, Heidelberg.
[17] Yang X S and Deb S 2009 December. Cuckoo search via Lévy flights *2009 World congress on nature & biologically inspired computing (NaBIC)* (pp. 210-214). IEEE.
[18] Singh N 2018 A modified variant of grey wolf optimizer *Int J Sci Technol Sci Iran* http://scientiairanica. sharif. edu.
[19] Singh N and Singh S B 2017 A modified mean gray wolf optimization approach for benchmark and biomedical problems *Evolutionary Bioinformatics* **13** 1176934317729413.
[20] Kumar A, Singh A and Kumar A 2019 December Weighted Mean Variant with Exponential Decay Function of Grey Wolf Optimizer on Applications of Classification and Function Approximation Dataset *In International Conference on Hybrid Intelligent Systems* pp. 277-290 Springer, Cham.
[21] Liang J, Suganthan P, and Deb K 2005 Novel composition test functions for numerical global optimization *In: Swarm intelligence symposium* SIS 2005 Proceedings 2005 IEEE; pp. 68–75.
[22] Sheskin D J 2020 Handbook of parametric and nonparametric statistical procedures *crc Press*.
[23] Storey C 1962 Applications of a hill climbing method of optimization *Chemical Engineering Science* **17**(1) pp. 45-52.
[24] Kumar A and Kumar A 2020 A Brief Literature on Optimization Techniques and Their Applications *In Innovations in Computer Science and Engineering* pp. 611-620 Springer, Singapore.
[25] Kumar A, Singh S and Kumar A 2021 Grey wolf optimizer and other metaheuristic optimization techniques with image processing as their applications: a review *In IOP Conference Series: Materials Science and Engineering* (Vol. 1136, No. 1, p. 012053) IOP Publishing.