

An omega tuned grey wolf optimization applied in binary classification problems

Akash Punhani [†]

Sumit Kumar ^{*}

Lalit Kumar [§]

School of Computer Science and Engineering

IILM University

Greater Noida 201306

Uttar Pradesh

India

Abstract

Popular and latest technology which is named as grey wolf optimizer (GWO) is very advantageous in hunting the grey wolves due to its meta heuristic behaviour of algorithm. In this algorithm the hunting process of the grey wolves have been defined based on the hierarchy of the wolves. However, in this hierarchical structure there has been no definition given to the wolves at the lowest level. These wolves are termed as omega wolves; they are termed to assist all the three categories of wolves. In this paper, we have modelled their role in the hunting process which was not considered in the previously proposed models. The definition for the contribution of Delta, Beta and Alpha wolves is modified in this approach. The proposed approach is tested on CEC 2017 benchmarks and proved to be more efficient in 12 out of 29 benchmarks for 30 Dimensions than its state of art variants. The proposed approach is used for classification with the help of multi-perceptron.

Subject Classification: 90C25, 90C29, 90C90.

Keywords: Omega tuned grey, Wolf, Classification.

[†] ORCID-ID: 0000-0002-4693-498X

^{*} ORCID-ID: 0000-0002-9494-1244

[§] ORCID-ID: 0000-0002-5132-4561

[†] E-mail: akashpunhani@gmail.com

^{*} E-mail: sumitnadar@gmail.com (Corresponding Author)

[§] E-mail: lalitkumar@iilm.edu

1. Introduction

In real life, we generally come across the problems for which either maximization or minimization of some real-life problem is to be done this may be represented in the form of the mathematical function. The name assigned to this function is objective function. The search for the optimal value for the objective function is termed as optimization problem. The most popular technique when we have complex functions whose solutions are difficult to be identified in polynomial time are solved using the heuristic algorithm inspired by nature.

Mirjalili, Mirjalili, and Lewis [1] proposed the essence inspired Meta heuristics-based maximization algorithm that imitates the social behaviour of the wolves while they hunt for the prey. As the wolves, follow the strict ranking, based upon the social status in the society. During the study, the wolves are classified into the four categories they are labelled as alpha, beta, delta and omega wolves. Fig. 1 describes complete hierarchical ordering of the wolves into four categories. Alpha grey wolves are called as leaders and are responsible for all decisions where they hunt will take place, where they will sleep and eat? The Beta wolf supports the Alpha wolves in taking the decisions. They are the second position in the hierarchy; they can replace the alpha wolf if they are not well. Similarly, delta grey wolves are placed third position in hierarchy from the top and assist both the top hierarchy wolves (β and α); they are also responsible for executing the instruction of the top levels and can take the assistance of the last grade of grey wolves that are named as omega grey wolves.

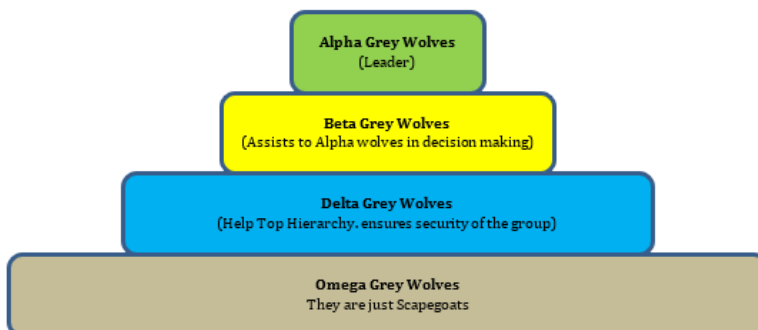


Figure 1
Hierarchical ordering of wolves in the study

The study presented in this paper is named as the optimization technique of grey wolf which is basically a latest algorithm of meta heuristic which perceives the nature of grey wolves' pack during predation process for achieving the favour of optimizing search. One thing is very noteworthy in this research is that there is a fixed hierarchy of grey wolves. According to this fixed hierarchy, grey wolf packs are divided into groups of four levels based on their physical and

societal status. The well-known four groups along with the hierarchal position are described in the Fig. 1. The highest position in the hierarchy is termed as the alpha wolves; they are at the highest position are generally rulers among all due to its best timeline while hunting. Second level holds the beta grey wolves because of their best decisions with other wolves. Third level in this strict hierarchy gives the position to those grey wolves which are best in obeying the decisions of upper two levels of wolves in hierarchy and they are termed as delta wolves. The remaining wolves that don't belong to the top hierarchy of those grey wolves are placed in the bottom most category and are termed as the omega wolves. This strict hierarchy of grey wolves plays an essential role during the hunting process. On the commands of alpha wolves, beta and delta chase and siege the prey and whenever prey tries to escape, the other wolves make the movements like wolf pack encirclement to catch that prey ultimately [1].

1.1 Encircling the Prey:

As the process of hunting of the prey begins the initially the locations of the prey are identified and are mathematically represented by the equations 1 and 2:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| d \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

In above equation, existing iteration is represented by symbol t , X_p is used to describe the coordinates of prey and X is used to represent the coordinates of wolf in population. The notation A and C . represent the turning parameter. The value of the A and C vectors are used in the computation which are coefficient vector.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2\vec{r}_2 \quad (4)$$

In the above equation the coefficient is generated by the two random numbers r_1 and r_2 . The value of the variable is reduced from 2 to 0 in a linear fashion.

1.2 Hunting the prey:

Once the encircling process is completed the next process is to hunt the prey. As the prior knowledge of the prey is not known hence the top three best population is termed as α , β , δ are selected as greatest known candidates for position of prey and dimension update is calculated by the equation 5(a-c):

$$GW\vec{D}_\alpha = |GW\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (5.a)$$

$$GW\vec{D}_\beta = |\vec{GW}\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (5.b)$$

$$GW\vec{D}_\delta = |\vec{GW}\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (5.c)$$

According to location of three variables, new location is calculated by equation 6(a-c) and 7

$$\vec{x}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\text{GW}\vec{D}_\alpha) \quad (6.a)$$

$$\vec{x}_2 = \vec{X}_\beta - \vec{A}_1 \cdot (\text{GW}\vec{D}_\beta) \quad (6.b)$$

$$\vec{x}_3 = \vec{X}_\delta - \vec{A}_1 \cdot (\text{GW}\vec{D}_\delta) \quad (6.c)$$

$$\vec{X}_{t+1} = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3} \quad (7)$$

Attacking the prey: The wolves will complete their procedure of hunting by combating the prey when it's still (not moving) is mentioned by equation (3).

Searching prey: Even though the operator provided helps in inception of the search space still there is the need of the exhaustive exploration of the search space this done they additional coefficient vector C that retain the randomness in the solutions as described by the equation (4).

2. Literature review:

The simplicity of the grey wolf optimizer and robustness has attracted the various researcher to work on grey wolf optimizer and has led to is generation of the various variant of the GWO.

Several improvements and hybrid variants of the Grey Wolf Optimizer (GWO) have been proposed to enhance its exploration, exploitation, and convergence performance. Saremi, Mirjalili, and Mirjalili [2] introduced evolutionary population dynamics to remove poor solutions and reposition new ones near better wolves, improving the overall fitness of the population but causing challenges such as premature convergence and reduced diversity.

Kamboj [3] proposed a hybrid PSO–GWO algorithm where particle positions are initially updated using a Novel PSO and later refined using GWO operators for solving the Unit Commitment Problem (UCP). Guha, Roy, and Banerjee [4] introduced the Quasi-Oppositional Grey Wolf Optimization (QOGWO) based on oppositional learning to accelerate convergence using quasi-oppositional numbers and a jumping rate parameter. Mittal, Singh, and Sohi [5] improved the algorithm by applying linear parameter adjustment to balance exploration and exploitation.

Jayabarathi et al. [6] proposed a hybrid GWO incorporating Differential Evolution (DE) operators such as mutation and crossover along with elitism, achieving improved performance for Economic Load Dispatch (ELD) problems. Malik, Mohideen, and Ali [7] modified the hunting mechanism using weighted averaging of wolf positions to improve convergence speed. Salahaldeen and Marwan [8] combined Genetic Algorithm (GA) and GWO, where GA performs initial exploration and its results initialize the GWO population.

Zhang, Luo, and Zhou [9] proposed Elite Opposition-based GWO (EOGWO) to enhance diversity and convergence speed, while Singh and Singh [10] hybridized GWO with the Sine Cosine Algorithm (SCA) to replace poor solutions and improve convergence. Song, Wang, and Lu [11] applied GWO with ensemble empirical mode decomposition for forecasting problems, improving prediction accuracy through preprocessing.

Long et al. [12] introduced Exploration-Enhanced GWO (EEGWO) using PSO-like updating and nonlinear control parameters to improve search performance. Gaidhane and Nigam [13] proposed a hybrid GWO–Artificial Bee Colony (ABC) approach with chaotic maps and opposition-based learning to avoid premature convergence. Panwar et al. [14] developed a binary GWO using sigmoid transfer functions and crossover operators for solving the Unit Commitment Problem.

Other variants include kernel optimization using modified wolf hierarchy [15], multi-strategy ensemble GWO for feature selection [16], dynamic dimensioned search GWO to avoid local minima [17], and hybrid GWO–Harmony Search for automatic parameter tuning [18]. Random opposition-based GWO and evolutionary operator-based GWO were also proposed to improve exploration capability [19–21].

Selective opposition-based GWO using Spearman correlation was proposed to enhance exploration in specific dimensions [22]. Binary GWO has also been successfully applied to feature selection problems, identifying a small subset of relevant features from larger datasets [31]. Additionally, an Improved GWO (I-GWO) based on Dimension Learning Hunting strategy allows wolves to learn from neighbours and retain the best update, demonstrating superior performance on CEC 2018 benchmark functions [30]. The various real world application of GWO and variants have been proposed by different researchers in [23–28] which makes the algorithm for solving complex real-world optimization problems.

3. Omega tuned grey wolf optimizer (OTGWO):

In this approach the four level of hierarchy, wolves have been involved. In this approach the omega are the worst wolves in the term of fitness in the complete pool of wolves. The enriching process is like that of the basic GWO approach however the hunting mechanism has been stated below:

3.1 Hunting the prey:

Once the encircling process is completed the next process is to hunt the prey. As the prior knowledge of the prey is not known hence the top three best population is termed as α , β , δ are selected as finest known candidates for position of prey and dimension update is calculated by the equation 8(a-c) and the worst population updates are defined by 8.d

$$GW\vec{D}_\alpha = |\overrightarrow{GWC}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (8.a)$$

$$GW\vec{D}_\beta = |\overrightarrow{GWC}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (8.b)$$

$$GW\vec{D}_\delta = |\overrightarrow{GWC}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (8.c)$$

$$GW\vec{D}_\Omega = |\overrightarrow{GWC}_4 \cdot \vec{X}_\Omega - \vec{X}| \quad (8.d)$$

According to location of three variables, new location is calculated by equation 9(a-c)

$$\vec{x}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (GW\vec{D}_\alpha) \quad (9.a)$$

$$\vec{x}_2 = \vec{X}_\beta - \vec{A}_1 \cdot (GW\vec{D}_\beta) \quad (9.b)$$

$$\vec{x}_3 = \vec{x}_\delta - \vec{A}_1 \cdot (\overrightarrow{GWD}_\delta) \quad (9.c)$$

The value of the new population is estimated using the representation as described by the condition in equation 10.

$$\vec{X}(t+1) = \left\{ \begin{array}{ll} \frac{\sum_{i=1}^3 \vec{x}_i}{3} & \text{if } \overrightarrow{GWD}_\omega < (\overrightarrow{GWD}_\alpha, \overrightarrow{GWD}_\beta, \overrightarrow{GWD}_\delta) \\ \frac{\vec{x}_2 + \vec{x}_3}{2} & \text{if } \overrightarrow{GWD}_\alpha < (\overrightarrow{GWD}_\omega, \overrightarrow{GWD}_\beta, \overrightarrow{GWD}_\delta) \\ \frac{\vec{x}_1 + \vec{x}_3}{2} & \text{if } \overrightarrow{GWD}_\beta < (\overrightarrow{GWD}_\alpha, \overrightarrow{GWD}_\omega, \overrightarrow{GWD}_\delta) \\ \frac{\vec{x}_1 + \vec{x}_2}{2} & \text{if } \overrightarrow{GWD}_\delta < (\overrightarrow{GWD}_\alpha, \overrightarrow{GWD}_\beta, \overrightarrow{GWD}_\omega) \end{array} \right\} \quad (10)$$

The above equation has the 4 cases:

1. If wolf is near to omega wolf, then it has to move according to the contribution of all the three top hierarchy that is alpha, beta and gamma.
2. If the wolf is near alpha, then it has to move according to the beta and gamma.
3. If the wolf is near beta, then it has to move according to the alpha and gamma.
4. If the wolf is near gamma, then it has to move according to the beta and alpha.

Algorithm 1: Omega Tuned GWO

Input: N, D, max_iter

Output: Global optimal values

Initialize starting population of N grey wolves randomly in area of search space and assign suitability

Repeat

 Identify $X_\alpha, X_\beta, X_\delta, X_\omega$

For every wolf w in N

 Change the location of wolf based on equation 10

End

Calculate a, A and C

Change $X_\alpha, X_\beta, X_\delta, X_\omega$

Val.t = val.t + 1

until (Iter < max_iter)

return X_α

4. Results and Discussions

The different algorithms are executed on the benchmark with fixed 3000 number of function evaluations, this means that GWO, OTGWO and MGWO will have exactly 3000 iterations on other hand the IIGWO will have only 1500 iterations, the reason behind doing so is that IIGWO algorithm consumes 2 levels one for GWO and other for DLH. Each algorithm is implemented for five rounds and their reported their mean and standard deviations.

Table 1
Performance of Algorithms on Unimodal Function

		GWO	OTGWO	IIGWO	MGWO
F1	Mean	707816937.34	891129349.10	10782065.61	1694144466.16
	STD	464303640.59	770874674.46	7046733.77	1089026222.89
F3	Mean	19145.87	8087.18	3759.01	41707.97
	STD	6813.56	2186.06	850.71	12027.01
	Gained Rank = (W-L-T)	0-2-0	0-2-0	2-0-0	0-2-0

In the table 1 the unimodal functions are compared and there mean and standard deviations are reported. From the results it can be observed that IIGWO has win in comparison to the other three algorithms.

Table 2
Performance of Algorithms on hybrid Function

		GWO	OTGWO	IIGWO	MGWO
F11	Mean	1362.35	1239.07	1276.65	2140.61
	STD	36.62	16.47	13.34	512.37
F12	Mean	33417456.59	22446639.21	41103691.12	98047637.20
	RANK	32195477.75	19482109.53	8526865.83	132678765.14
F13	Mean	70155.63	883869.53	5219025.96	87344.68
	STD	28741.45	1558948.09	4270504.88	47842.68
F14	Mean	258847.31	35380.93	14306.12	232804.96
	STD	384285.10	21006.18	7391.18	245557.12
F15	Mean	44456.72	34696.83	219753.54	456068.98
	STD	19872.00	22689.68	76098.07	938481.45
F16	Mean	2584.13	2053.10	2421.99	2382.46
	STD	174.96	172.23	161.43	198.27
F17	Mean	2008.02	1938.90	1905.44	1966.88
	STD	207.14	130.01	88.78	121.12
F18	Mean	392834.69	323049.02	504010.19	1594428.05
	STD	494808.73	231578.94	183889.46	1720689.82
F19	Mean	256965.33	1856822.07	863419.62	264052.31
	STD	420388.96	3937880.64	648519.33	370891.86
F20	Mean	2281.84	2315.33	2281.26	2383.41
	STD	90.93	214.59	102.39	52.01
	Gained Rank = (W-L-T)	2-8-0	5-5-0	3-7-0	0-0-0

Table 2 presents the outcomes achieved following the application of four algorithms to ten hybrid functions from cec2017. Notably, OTGWO demonstrated superior performance on five of these functions compared to the other three optimization algorithms.

Table 3
Performance of Algorithms on composite Function

		GWO	OTGWO	IIGWO	MGWO
F21	Mean	2368.05	2371.27	2468.96	2390.72
	STD	8.23	13.25	10.85	17.32
F22	Mean	4363.40	6655.31	3575.63	4329.86
	STD	1745.03	2538.49	2828.86	1428.88
F23	Mean	2727.20	2702.54	2817.74	2783.09
	STD	11.78	12.42	16.19	28.45
F24	Mean	2912.35	2950.00	2992.79	2968.45
	STD	61.79	79.06	4.46	50.38
F25	Mean	2944.54	2937.69	2892.13	2993.23
	STD	11.61	17.13	3.50	54.28
F26	Mean	4500.38	4361.19	4846.51	4630.65
	STD	325.99	164.37	1143.27	677.92
F27	Mean	3230.93	3220.99	3230.11	3269.71
	STD	6.18	4.89	9.15	28.72
F28	Mean	3350.39	3352.92	3253.67	3427.08
	STD	65.14	41.31	14.12	40.22
F29	Mean	3667.97	3602.82	3693.90	3785.72
	STD	95.09	113.38	62.43	214.27
F30	Mean	4427404.77	4544211.65	2709467.57	6320347.46
	STD	2835003.49	3863121.84	1308468.37	2887040.28
	Gained Rank = (Win-Loose-Tie)	2-8-0	4-6-0	4-6-0	0-0-0

Table 3 represents the results obtained on the execution of algorithms on hybrid function both IGWO and GWO can reach near to optimal solutions whereas GWO also get the near optimal solution.

Overall Effectiveness

The results for all four types of functions in the CEC2017 benchmarks have been obtained. The combined results for the complete benchmark are presented in Table 4 below. It is observed that OTGWO algorithm is best in comparison among all three algorithms.

Table 4
Represents the overall comparison on 29 benchmark functions

Optimizer	RANK	Percentage
GWO	6/23/0	20.68965517
OTGWO	12/17/0	41.37931034
IIGWO	11/18/0	37.93103448
MGWO	0/29/0	0

Statistical Evaluation:

To prove the excellence of the algorithm under discussion that is OTGWO a non-parametric test was used. Data of several Friedman ranks are shown in below table 5. In this problem the k is the number of algorithms that are under consideration as we have taken 4 algorithms so $k = 4$, the n is equal to the number of functions that is 29.

The mathematical equation as described by equation 11, based the values obtained we can clearly state the for the significance level value p is assumed to be 0.05. The p values are less than the desired level.

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^k R_j^2 - 3n(k-1) \quad (11)$$

Table 5
Represents the Friedman ranks and overall ranks

FUNCTIONS	GWO	OTGWO	IIGWO	MGWO
F1	2.8	2.8	1	3.4
F3	3	2	1	4
F4	2.4	2.4	1.4	3.8
F5	2	1.6	3.8	2.6
F6	2	1.8	2.2	4
F7	2.4	1.8	3.4	2.4
F8	1.8	1.8	3.8	2.6
F9	3	2.6	1	3.4
F10	1.8	2.2	4	2
F11	3	1.2	1.8	4
F12	2.2	1.4	3	3.4
F13	2	2.4	3.8	1.8
F14	2.6	2.2	1.8	3.4
F15	2.2	1.8	3.8	2.2
F16	3.4	1.4	2.6	2.6
F17	2.6	2.6	1.8	3
F18	1.8	1.6	2.8	3.8
F19	2	2.4	3.8	1.8
F20	2.4	2.6	2.2	2.8
F21	1.4	1.8	4	2.8
F22	2.6	3.6	1.4	2.4
F23	1.8	1.2	3.6	3.4
F24	1.8	2.4	3	2.8
F25	3	2.8	1	3.2
F26	2	1.8	3.2	3
F27	2.8	1.2	2.2	3.8
F28	2.4	2.8	1	3.8
F29	2.2	1.8	3	3
F30	2.6	2.4	2	3
Avg. Rank	2.344828	2.082759	2.531034	3.04137931
Overall RANK	2	1	3	4
SUM of RANK	68	60.4	73.4	88.2
SQ. SUM of RANK	4624	3648.16	5387.56	7779.24

5. Application

To analyse the performance of the OTGWO algorithm, the have applied proposed algorithm to binary classification problem on the five dataset that can be accessed from the UCI repository (archive.ics.uci.edu/ml/datasets.php), all these datasets were employed to train neural network with the help of optimizer proposed by [29]. For the executing the model the population is 10 and 5 run of the same were performed and the 40 iterations for each of the algorithm were performed. Each of the dataset is separated into two portions, one is 66% of data which is used in training and remaining is 34% which is used in testing. Other results of GWO, OTGWO are comparable, they are better for diabetes and liver dataset but not good for Parkinson and vertebral dataset are presented in the figure 2.

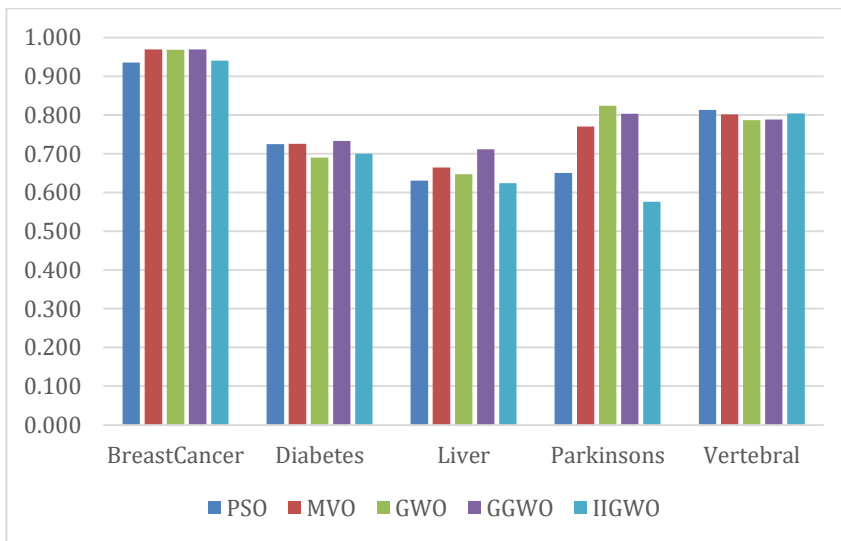


Figure 2
Describes the accuracy on different datasets

6. Conclusion

The proposed OTGWO is better alternative to IIGWO as the results obtained for OTGWO are better in comparison to IIGWO. Secondly the IIGWO required more iterations, and the algorithm is having complex computations. The proposed variant of GWO is giving the role to omega wolves that are ignored until now. According to the result of cec2017 the results are better in 12 of benchmark function also in the case of the application to the classification problem we have got good results in 3/5 cases. As it has been stated by the various researcher that there is no single algorithm which can be better in all aspects and it is problem specific. In future, we are focus on other aspect of GWO algorithm like learning rate and oppositional techniques.

References

- [1] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61 (2014).
- [2] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and Grey Wolf Optimizer," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1257–1263 (2015).
- [3] V. K. Kamboj, "A novel hybrid PSO–GWO approach for unit commitment problem," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1643–1655 (2016).
- [4] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of large-scale power system using quasi-oppositional Grey Wolf Optimization algorithm," *Engineering Science and Technology, an International Journal*, vol. 19, no. 4, pp. 1693–1713 (2016).
- [5] N. Mittal, U. Singh, and B. S. Sohi, "Modified Grey Wolf Optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing*, vol. 2016, pp. 1–16 (2016).
- [6] T. Jayabarathi, T. Raghunathan, B. R. Adarsh, and P. N. Suganthan, "Economic dispatch using hybrid Grey Wolf Optimizer," *Energy*, vol. 111, pp. 630–641 (2016).
- [7] M. R. S. Malik, E. R. Mohideen, and L. Ali, "Weighted distance Grey Wolf Optimizer for global optimization problems," in *Proc. IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–6 (2015).
- [8] J. Salahaldeen and M. Marwan, "The use of original and hybrid Grey Wolf Optimizer in estimating the parameters of software reliability growth models," *International Journal of Computer Applications*, vol. 167, no. 3, pp. 12–21 (2017).
- [9] S. Zhang, Q. Luo, and Y. Zhou, "Hybrid Grey Wolf Optimizer using elite opposition-based learning strategy and simplex method," *International Journal of Computational Intelligence and Applications*, vol. 16, no. 2, Art. no. 1750014 (2017).
- [10] N. Singh and S. B. Singh, "A novel hybrid GWO–SCA approach for optimization problems," *Engineering Science and Technology, an International Journal*, vol. 20, no. 6, pp. 1586–1601 (2017).
- [11] J. Song, J. Wang, and H. Lu, "A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting," *Applied Energy*, vol. 215, pp. 643–658 (2018).

- [12] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced Grey Wolf Optimizer to solve high-dimensional numerical optimization," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 63–80 (2018).
- [13] P. J. Gaidhane and M. J. Nigam, "A hybrid Grey Wolf Optimizer and artificial bee colony algorithm for enhancing the performance of complex systems," *Journal of Computational Science*, vol. 27, pp. 284–302 (2018).
- [14] L. K. Panwar, K. S. Reddy, A. Verma, B. K. Panigrahi, and R. Kumar, "Binary Grey Wolf Optimizer for large-scale unit commitment problem," *Swarm and Evolutionary Computation*, vol. 38, pp. 251–266 (2018).
- [15] Cai, Zhennao, Jianhua Gu, Jie Luo, Qian Zhang, Huiling Chen, Zhifang Pan, Yuping Li, and Chengye Li, "Evolving an optimal kernel extreme learning machine by using an enhanced Grey Wolf Optimization strategy," *Expert Systems with Applications*, vol. 138, Art. no. 112814 (2019).
- [16] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble Grey Wolf Optimizer and its application to feature selection," *Applied Soft Computing*, vol. 76, pp. 16–30 (2019).
- [17] F. Yan, J. Xu, and K. Yun, "Dynamically dimensioned search Grey Wolf Optimizer based on positional interaction information," *Complexity*, vol. 2019, Art. no. 6521268 (2019).
- [18] A. A. Alomoush, A. A. Alsewari, H. S. Alamri, K. Aloufi, and K. Z. Zamli, "Hybrid harmony search algorithm with Grey Wolf Optimizer and modified opposition-based learning," *IEEE Access*, vol. 7, pp. 68764–68785 (2019).
- [19] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, "A random opposition-based learning Grey Wolf Optimizer," *IEEE Access*, vol. 7, pp. 113810–113825 (2019).
- [20] A. Saxena, R. Kumar, and S. Mirjalili, "A harmonic estimator design with evolutionary operators equipped Grey Wolf Optimizer," *Expert Systems with Applications*, vol. 145, Art. no. 113125 (2020).
- [21] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved Grey Wolf Optimizer for solving engineering problems," *Expert Systems with Applications*, vol. 166, Art. no. 113917 (2021).

- [22] S. Dhargupta, M. Ghosh, S. Mirjalili, and R. Sarkar, "Selective opposition-based Grey Wolf Optimization," *Expert Systems with Applications*, vol. 151, Art. no. 113389 (2020).
- [23] X. Song et al., "Grey Wolf Optimizer for parameter estimation in surface waves," *Soil Dynamics and Earthquake Engineering*, vol. 75, pp. 147–157 (2015).
- [24] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman, "Using the Grey Wolf Optimizer for solving optimal reactive power dispatch problem," *Applied Soft Computing*, vol. 32, pp. 286–292 (2015).
- [25] S. Mirjalili, "How effective is the Grey Wolf Optimizer in training multi-layer perceptrons," *Applied Intelligence*, vol. 43, no. 1, pp. 150–161 (2015).
- [26] M. Pradhan, P. K. Roy, and T. Pal, "Grey Wolf Optimization applied to economic load dispatch problems," *International Journal of Electrical Power & Energy Systems*, vol. 83, pp. 325–334 (2016).
- [27] N. Singh and S. B. Singh, "A modified mean Grey Wolf Optimization approach for benchmark and biomedical problems," *Evolutionary Bioinformatics*, vol. 13, pp. 1–20 (2017).
- [28] R. Katarya and O. P. Verma, "Recommender system with Grey Wolf Optimizer and FCM," *Neural Computing and Applications*, vol. 30, no. 5, pp. 1679–1687 (2018).
- [29] H. Faris, I. Aljarah, S. Mirjalili, P. Castillo, and J. J. Merelo, "EvoPy: An open-source nature-inspired optimization framework in Python," in *Proc. 8th International Joint Conference on Computational Intelligence (ECTA)*, pp. 171–177 (2016).
- [30] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, pp. 372–379 (2017).
- [31] J. Mante and K. Kolhe, "A novel DDoS detection framework leveraging hybrid binary Grey Wolf Optimization and explainable AI," *Journal of Information and Optimization Sciences*, vol. 46, no. 2, pp. 563–578 (2025).