Wasit Journal for Pure Science

Journal Homepage: https://wjps.uowasit.edu.iq/index.php/wjps/index
e-ISSN: 2790-5241 p-ISSN: 2790-5233

Eight-Figure Pattern for Enhancing the Searching Process of Grey Wolf Optimization (Eight-GWO)

Ali Hakmem Alsaeedi[1,2], Suha Muhammed Hadi[1], Yarub Alazzawi[3]*, Emad Badry[4]

[1]Informatics Institute for Postgraduate Studies University of Information Technology and Communications, IRAQ

[3]College of Computer Science and Information Technology, University of Al-Qadisiyah, IRAQ

[2]Al-Khwarizmi College of Engineering, University of Baghdad, IRAQ
[4]Department of Electrical Engineering, of Engineering, Suez Canal University, Ismailia, EGYPT

*Corresponding Author: Ali Hakem Alsaeedi

DOI: https://doi.org/10.31185/wjps.718
Received 10 January 2025; Accepted 09 March 2025; Available online 30 Jun 2025

ABSTRACT: Evolutionary algorithms suffer significantly from stagnation at local optima. This paper proposes a new strategy that detects when the search gets stuck in a local optimum and then adopts a dynamic search strategy to escape stagnation. The proposed model is based on simulating eight pattern movements and embedded with a Grey Wolf Optimizer algorithm (GWO). It is called the Eight-Figure Grey Wolf Optimizer (Eight-GWO). The proposed model combines two phases: regular search when searching progresses over time while the second phase, searching by eight patterns when the algorithm reaches stuck. The Eight-pattern updates the gray position based on the sin and cos function. The Eight-pattern updates the gray position using sine and cosine functions to introduce controlled oscillations in the search process. These trigonometric functions help balance exploration and exploitation by dynamically periodically adjusting positions. The proposed Eight-GWO on the 24 functions of the CEC2005 benchmark suite and compared its results with both the standard GWO and Particle Swarm Optimization (PSO). The experiments result show the proposed Eight-GWO gets better results than GWO and PSO where it achieved the best results on 80% of the test functions. The proposed Eight-GWO runs 23% faster than the original GWO and 44% faster than PSO.

Keywords: Evolutionary algorithms, Grey Wolf Optimizer, Stagnation, Random search, stochastic exploration

1. INTRODUCTION

In recent years, metaheuristic optimization algorithms have been used to solve several complex problems[1], such as those in [2], Agriculture [3], networks[4-6], medicine [1, 7, 8], etc. The fundamentals of these algorithms are based on the stochastic exploration of the searching problem. It employs systematic random search to control the random value produced by the optimizer generator[9]. To perform a systematic random search, metaheuristic techniques must fully explore the search space while guiding the search toward a global or near-global optimum. The exploration required a random search and variety in generating candidate solutions[10-12]. This involves generating a diverse array of candidate solutions through random sampling. Additionally, when the algorithm identifies the best candidate solution, it incorporates it into the current pool of candidates, further enhancing the exploration process.

Several evolutionary algorithms exist, most inspired by the natural behavior of animals and phenomena observed in nature[1]. For example, Particle Swarm Optimization (PSO), Bat Optimization Algorithm (BOA), Grey Wolf Optimization (GWO), … so on. Mirjalili et al. [7] introduced Grey Wolf Optimization (GWO) in 2014. This algorithm was inspired by the hunting of gray wolves. The GWO algorithm has been successfully applied across various fields [13].

 **Corresponding author:* ali.alsaeedi@qu.edu.iq

Generally, evolutionary algorithms suffer from a lack of convergence [8-10]. As a result, they can get stuck in local optima early in the search progress [1, 14]. These challenges often reduce effectiveness in addressing real-world optimization problems. Therefore, they required efficient search strategies that enhance both exploration and exploitation.

The standard Grey Wolf Optimizer (GWO) faces several limitations, particularly in handling complex optimization problems. One of the most critical issues is stagnation, which occurs when the algorithm struggles to escape local optima, leading to premature convergence and suboptimal solutions.

In GWO, the hunting mechanism is guided by the alpha, beta, and delta wolves, which update positions based on their relative dominance. However, as iterations progress, the search agents tend to converge around these leading solutions, reducing diversity in exploration. This limits the ability to discover better solutions, especially in multimodal landscapes.

Stagnation significantly affects the algorithm's efficiency in solving high-dimensional and complex optimization problems where numerous local optima exist. It restricts the ability to explore new regions, leading to low convergence speed and poor adaptability to dynamic environments. To overcome stagnation, enhanced versions of GWO incorporate adaptive mechanisms, chaotic maps, or hybrid strategies to maintain diversity and improve global search capability.

This paper proposed an enhanced version of the Grey Wolf Optimizer embedded with an Eight-Figure pattern. This pattern helps prevent stagnation and improves the algorithm's exploration capabilities, thus enhancing its performance in complex optimization problems. Technically, the proposed model incorporates an adaptive strategy that detects stagnation and switches to a more dynamic search pattern when the algorithm gets trapped in local optima. The proposed Eight-Figure same infinity symbol (∞). Therefore, this movement can be represented mathematically in combination with two trigonometric equations, sine and cosine. A geometrically inspired strategy, such as the eight-figure pattern, offers a novel yet structurally simple mechanism to diversify movement patterns without sacrificing the algorithm's original elegance or efficiency. The symmetry and directional properties of the eight-figure trajectory inherently promote systematic exploration while preserving the social hierarchy's exploitation benefits.

1.1 Motivation

The development of the Eight-GWO algorithm is driven by several critical motivations rooted in both theoretical and practical challenges observed in the standard Grey Wolf Optimization (GWO) algorithm. While GWO has proven effective in various optimization scenarios, its performance often degrades when applied to complex, multimodal, or high-dimensional problems. These limitations stem from inherent algorithmic behaviors that restrict its adaptability and scalability in dynamic search environments.

1. **Limit exploration**: The standard GWO heavily relies on the hierarchical leadership structure (alpha, beta, delta wolves) to guide the search process. While this mechanism ensures the exploitation of the best solutions, it inherently limits exploration diversity, especially in the early stages of optimization.
2. **Complexity**: Existing modifications to GWO, such as hybridization with chaos theory, Lévy flights[15], PSO[16], or other optimization algorithms, have shown partial success in mitigating stagnation. However, many of these approaches introduce computational overhead or hyperparameters that complicate implementation and reduce reproducibility.
3. **Stuck in local optima**: the standard GWO is often stuck in local optima due to a leak in balancing between the exploration and exploitation.

1.2 Contributions

The contributions of this work are threefold:

1- The conceptualization and mathematical formulation of the eight-figure pattern to enhance GWO's search capabilities.
2- Reducing complexity in the proposed model by changing the search direction of the algorithm to a technique with fewer parameters and higher exploration.
3- Enhancing the exploration and exploration in the proposed search strategy.

1.3 Evaluation strategy

Optimization algorithms rely on randomness during the search process, which may lead to variations in results when repeated[17]. Therefore, to evaluate the algorithm's efficiency must be executed multiple times independently[17]. The algorithm's performance is calculated from these executions: average, best, worst, median, and the standard deviation of best solutions. Additionally, boxplot charts and Convergence plots visually depict the algorithm's performance[18]. The boxplot charts can be used to visualize the statistical distribution of results, highlighting the variability and potential outliers in the solutions. Convergence plots are also essential for analyzing how quickly and efficiently the algorithm approaches optimal solutions, providing further insight into its effectiveness.

1.4 Paper organization

The remainder of this paper is organized as follows: Section 2 related works and Section 3 provides a brief overview of the standard GWO algorithm. Section 4 details the design and implementation of the proposed Eight-GWO framework. Section 5 presents experimental results and comparative analyses, while Section 5 discusses practical applications and implications. Finally, Section 7 concludes the study and outlines future research directions.

2. RELATED WORK

Various strategies have been used to improve the performance of the Gray Wolf Optimizer. Each work provides unique insights into mitigating specific limitations of the standard GWO, paving the way for further research and development in this area.

Liu et al. (2023) [15] proposed an enhanced strategy of GWO. They Combine Gaussian chaotic mapping for population initialization, a nonlinear convergence factor, Lévy flight for global exploration, and the golden sine algorithm for local exploitation. Enhances position updates dynamically to balance exploration and exploitation. Premature convergence due to poor population diversity and imbalanced exploration-exploitation phases in the original GWO. Moreover, the increased computational complexity may hinder performance for large-scale problems.

Salgotra et al. (2020) [15] introduced opposition-based learning during initial iterations to diversify search agents and assign diverse positions to leader wolves (alpha, beta, delta) to expand exploration. This method was validated using benchmark functions and antenna array design. Its limitation is poor exploration capability and susceptibility to local optima stagnation in the standard GWO.

In (2024) [19] Integrates GWO with Hybrid Rice Optimization (HRO) using dynamic parameter regulation, neighborhood search, dual-crossover, and selling techniques. A hybrid filter-wrapper framework with chi-square filtering improves feature selection accuracy. Limited adaptability, poor population diversity, and low accuracy in high-dimensional feature selection tasks. The hybrid approach increases algorithmic complexity due to the integration of two metaheuristics. This often requires additional parameter tuning, which may limit its applicability to certain problem domains.

Mohammed et al. [20] (2024) Modified GWO using gamma wolves, z-position updates, and the golden ratio to refine search trajectories. Evaluated CEC2019 benchmark functions and engineering problems (e.g., pressure vessel design). Local optima entrapment and low exploration efficiency in complex landscapes. The success of the adaptive mechanism depends on carefully calibrating control parameters. It may not generalize well across all types of optimization problems without further modifications.

Adegboye et al. [17] (2024) integrate chaotic opposition learning (COL), mirror reflection strategy (MRS), and worst individual disturbance (WID). MRS expands the exploration range, COL enhances diversification, and WID facilitates escape from local optima. Population diversity loss and stagnation in local optima due to over-reliance on alpha wolves. The algorithm becomes more complex with the integration of multiple strategies. Additionally, its performance gains are primarily observed in high-dimensional scenarios and may not be as effective for low-dimensional problems.

3. SEARCH MECHANISM OF METAHEURISTICS

The metaheuristic optimization search mechanism involves exploration and exploitation[10, 19, 21]. Exploration discovers new solutions based on randomness. At the same time, exploitation refines promising ones based on the best current solutions[4]. The balance between exploration and exploitation reduces scattering and leads to a more systematic search process for optimal solutions[22-24]. This balance is crucial in avoiding premature convergence and finding high-quality solutions in complex optimization landscapes.

Equation 1 illustrates the basic mathematical representation of the search process in metaheuristic optimization.

$$f(x') = f(x) + f(y) \qquad (1)$$

Where f(x) represents the current solution, and f(y) is an optimization value added to generate a new solution f(x′). As mentioned above, the optimization value composites of exploration f(r) and exploitation f(b). Equation 1 is the decomposition as follows.

$$f(x') = f(x) + (f(r) + f(b)) \qquad (2)$$

Adding a control function f(c) to govern the search process in metaheuristic algorithms is possible. This function can be a constant ($f(c_1)$) or derived from candidate solutions. Therefore, Equation 3 represents the final formula for the search process in metaheuristic optimization.

$$f(x') = f(c_1) + f(x) + (f(r) + f(b)) + f(c_2) \qquad (3)$$

4. GREY WOLF OPTIMIZATION

One of the famous optimization algorithms is the Grey Wolf Optimizer (GWO)[25]. It draws on the leader-follower behavior of grey wolves and their hunting strategy. The GWO categorizes wolves into four categories: alpha (α), beta (β), delta (δ), and omega (ω), among which α, β, and δ denote the top three solutions and omega wolves spend time guiding these top wolves (shown in figure1).

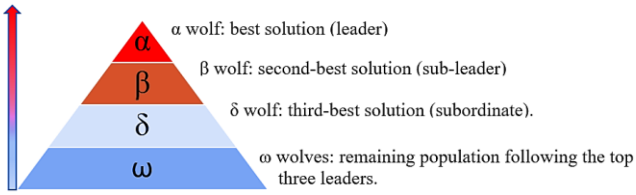

***FIGURE 1:* social Hierarchy of GW**

The GWO algorithm mathematically models the behavior of wolves encircling their prey[1]. Equation 5 represents the process of moving wolves toward optimal solutions:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{prey} - \vec{X}(t)| \quad (4)$$

$$\vec{X}(t+1) = \vec{X}_{prey} - \vec{A} \cdot \vec{D} \quad (5)$$

Where: $\vec{X}_{\text{prey}}$ is the position vector of the prey (target solution), $\vec{X}(t)$ is the position vector of a wolf at iteration t.and $\vec{A}$ and $\vec{C}$ are coefficient vectors which are calculated in equations 6 and 7, respectively:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r_1} - \vec{a} \quad (6)$$

$$\vec{C} = 2 \cdot \vec{r_2} \quad (7)$$

Where: $\vec{a}$ decreases linearly from 2 to 0 throughout iterations, and $\vec{r_1}$ and $\vec{r_2}$ are random vectors in the range $[0,1]$. The positions of α, β, and δ wolves are known, and other wolves update their positions based on these top wolves to converge toward the prey. The position of a new wolf is updated in equation 8:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (8)$$

Where: $\vec{X}_1$,$\vec{X}_2$,$\vec{X}_3$calculated in equations9,10, and 11, respectively.

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (9)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (10)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (11)$$

The GWO algorithm used a coefficient vector $(\vec{A})$ to balance between exploitation and exploration. When $|\vec{A}| > 1$, Wolves spread out, promoting exploration of the search space. This broad search helps the algorithm avoid premature convergence to local optima and improves the chances of finding the global optimum[26]. Conversely, when $|\vec{A}| < 1$, Wolves converge toward each other, focusing on refining the search around the most promising areas identified by the top solutions (α, β, and δ). This convergence enhances exploitation by allowing the algorithm to focus on searching within promising regions to refine solution quality[27]. Moreover, maintaining a dynamic balance between exploration and exploitation is essential for effectively navigating complex optimization landscapes.

Figure 2 illustrates the architecture of the search process in the Grey Wolf Optimizer (GWO)

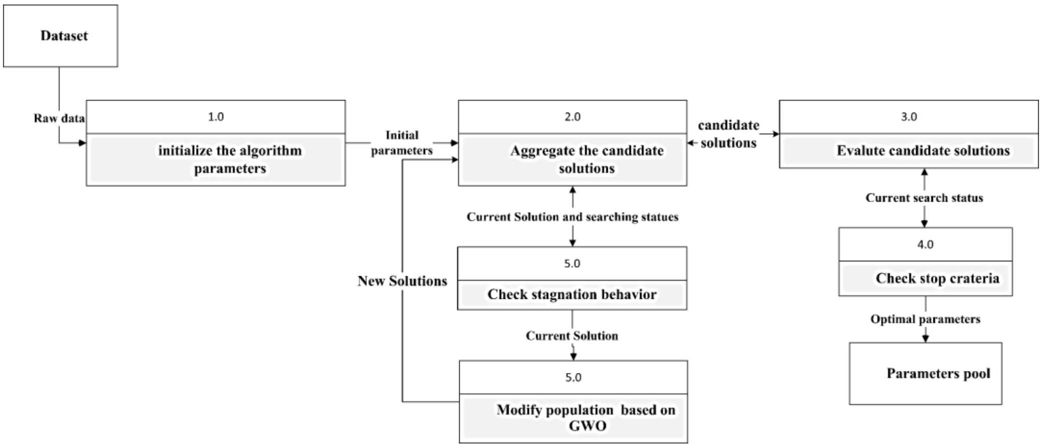

FIGURE 2: Architecture of the search process in the GWO

5. EIGHT-FIGURE GREY WOLF OPTIMIZATION (EIGHT-GWO)

Evolutionary algorithms (EAs) usually comprise three main phases: Initialization, iterative position update, and termination. The proposed eight-GWO algorithm improves this process by introducing a new search strategy that detects stagnation and dynamically adjusts the search behavior to avoid getting stuck in local optima, which improves exploration capabilities. Furthermore, this strategy addresses the challenge of search space dispersion by constraining the dimensionality of the wolf population without changing the established search patterns. Figure 3 illustrates the proposed architecture of Eight-GWO.

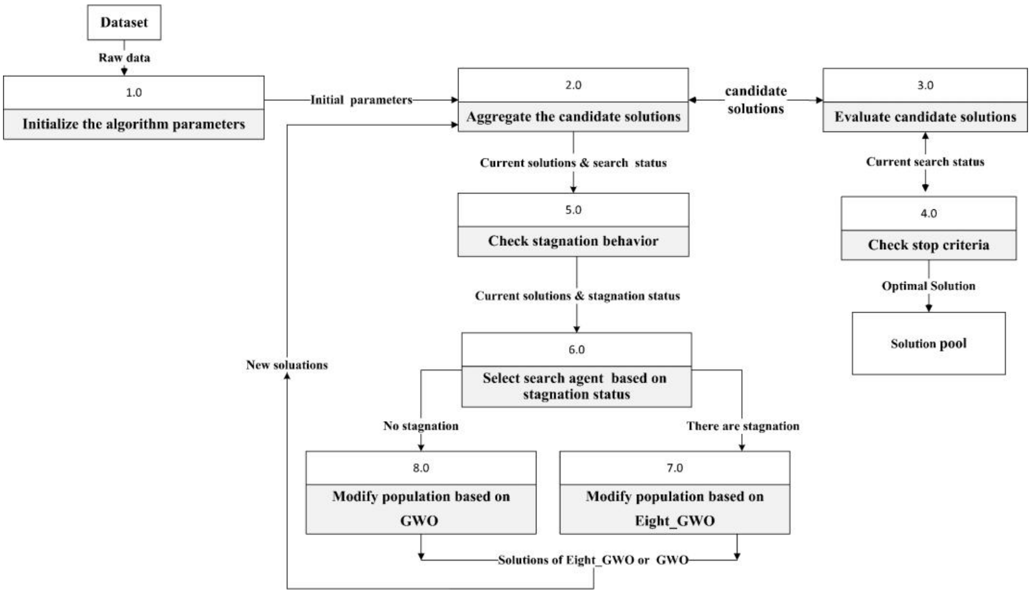

FIGURE 3: The architecture of the Proposed Eight-Figure Grey Wolf Optimization (Eight-GWO)

5.1 Initialization Eight_GWO algorithm:

The proposed Eight-GWO's initial parameters are carefully selected to balance exploration and exploitation while promoting algorithm sensitivity to stagnation. The major parameters are population size (P), minimum and maximum dimension updates (Dmin , Dmax), max_iteration (tmax), and maximum stagnation (Smax).

- **Population size (P):**

  The population size, a critical factor influencing the algorithm's diversity and search capability, is typically bounded between a minimum and maximum value. This range ensures sufficient exploration while preventing excessive computational overhead.

- **Minimum and Maximum dimension updates (Dmin , Dmax):**

  The dimension updates governing the wolves' movement within the search space are also fine-tuned to facilitate efficient exploration and convergence. Maximum stagnation (Smax) aims to maintain the algorithm's responsiveness to changes in the search landscape.

Eight-GWO avoids premature convergence by prioritizing sensitivity and effectively adapts to complex optimization problems. The careful selection of these initial parameters contributes to Eight-GWO's overall performance and robustness, making it a promising candidate for solving challenging optimization tasks across various domains.

5.2 Solutions aggregation and checking searching behavior:

This step comprises processes 2.0 to 6.0. It includes compiling the proposed solutions and monitoring the progress of the search. In process 3.0, the proposed solutions are evaluated by applying the fitness function. The proposed algorithm then performs two checks: the stopping criterion and stagnation. The stopping criterion is based on the maximum number of iterations. When this limit is reached, the algorithm terminates. The algorithm is also sensitive to stagnation if the best solution (alpha) remains unchanged for several iterations equal to 'Smax'; a stagnation flag is set, which prompts the algorithm to explore the search space with the proposed eight-digit pattern.

5.3 Select a search agent based on stagnation status:

This proposed algorithmic step guides the search engine to the GWO (Grey Wolf Optimizer) or the eight-figure (EF) pattern. The decision hinges on the algorithm's stagnation state. If stagnation is absent or its duration remains within the allowed limit (Smax), the algorithm proceeds with the grey wolf method for exploration. Conversely, if stagnation persists beyond the allowed period, the algorithm switches its search strategy to the proposed EF method, aiming to break free from the local optimum and enhance exploration. Figure 4 illustrates the proposed scenario when the searching process gets stuck at local optima.

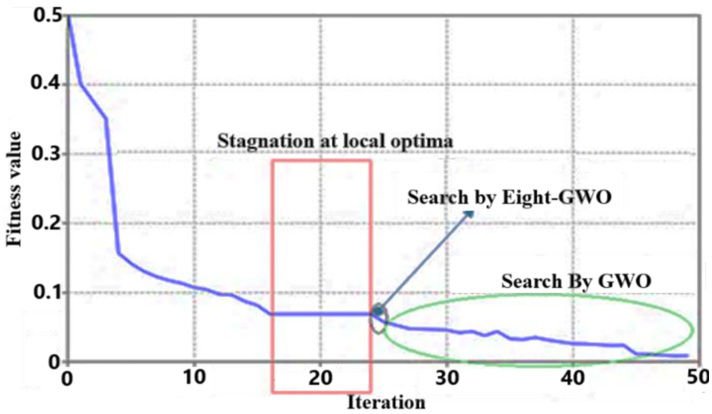

FIGURE 4: The proposed scenario of the search by EF_GWO

5.4 Modify the population based on Eight Figure (Eight figure) process

When the algorithm's searching process becomes trapped in local optima, the proposed algorithm enhances exploration using a new eight-figure pattern. It was inspired by a wolf's movement in a cage, as shown in Figure 5.

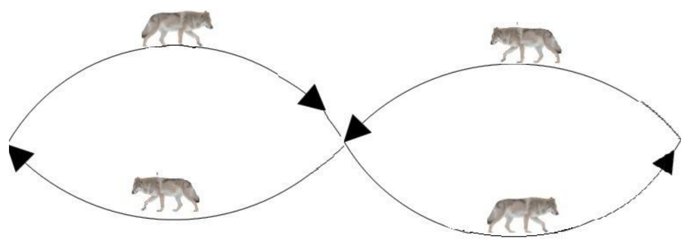

FIGURE 5: Movement in Eight figure pattern of a wolf

This circular motion can amplify power and range, especially within confined spaces. This movement pattern can be mathematically represented by parametric equations utilizing sine and cosine functions. The Eight-Figure search strategy for modifying the population comprises two stages: the determination of the wolf dominant from the updated population(wolves) and the application of a position modification function:

5.5 Determine wolf dimensions

To mitigate scattering in the early stages of the proposed EF framework's search process, a subset of dimensions is selected for modification at each iteration rather than adjusting the positions of all wolves across all dimensions. The probability of selecting more dimensions for modification increases as the number of iterations grows. This approach balances maintaining diversity and exploiting promising regions, dynamically adapting the search strategy to the evolving landscape of the optimization problem. The dimensions (WDim) of the wolf that are updated at iteration (t) are calculated using Equation 12.

$$W_{Dim} =\sim \left(\left(D_{min} \; + \left(\frac{t}{t_{max}}\right)^2 * (D_{max} - D_{min})\right) * D_p\right) \qquad (12)$$

Where: t is the current iteration, and D_p is a problem dimension. Figure 5 illustrates how the W_{Dim} increase during search progress.

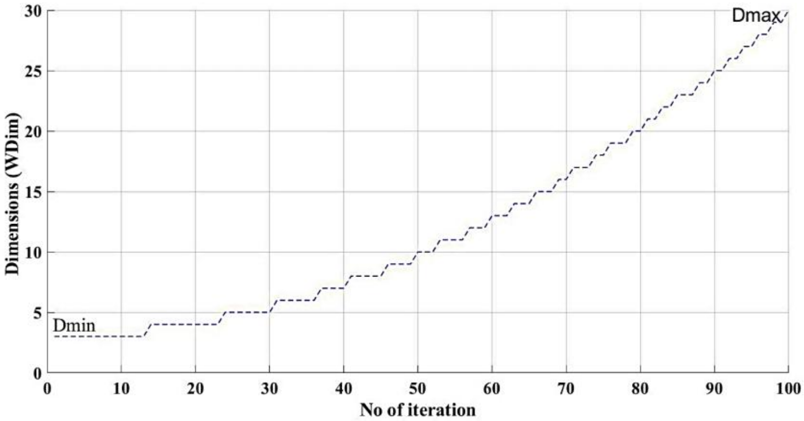

FIGURE 6: ***Example shows the relation between W and no. of iteration, in this figure, the set the D_{max} = 30, D_{min} = 5 , t_{max} = 100***

5.6 Update wolf position

To update the wolf's position and simulate the characteristic 'figure-eight' movement pattern observed in grey wolves' skipping from obstacle behavior, we suggest using Equation 13. This formula contains sine and cosine functions, allowing for a more nuanced representation of the wolves' encirclement and approach maneuvers during optimization.

$$X_{t+1} = r_1 \cdot X_\alpha + (1 - \Delta_{cos}) \cdot \Delta_{sin} \qquad (13)$$

Where : r_1 : Random value within the interval [0, 1]. X_α is a positions of the alpha wolf.

Equation 14 calculates the value of The Δ_{cos}

$$\Delta_{cos} = cos(2\pi L r_2) \cdot (r_3 + X_\alpha - X_\beta) \qquad (14)$$

Where: r_1, r_2: Random values within the interval [0, 1]. L is the Frequency or scaling factor. , X_β is the position of the beta wolves.

Equation 15 calculates the value of The Δ_{sin}

$$\Delta_{sin} = sin(2\pi L r_4) \cdot (r_5 + X_\beta + X_\delta) \qquad (15)$$

Where: r_4, r_5: random values within the interval [0, 1], X_δ is positions of the Sigma Wolf.

Figure 3.8 shows the implantation of functions equations 14 and 15.

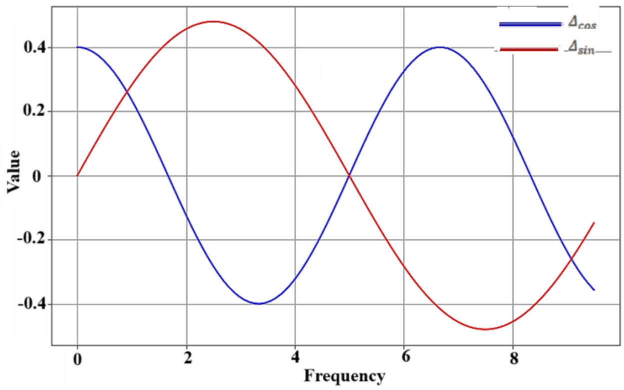

FIGURE 7: Example shows the implantation of Eq. (3. 19, and 3.20), the value of $r_2 = 0.15$, $r_3 = 0.2$, $r_4 = 0.1$, $r_5 = 0.18$, *and* $X_\alpha = 0.3$, $X_\beta = 0.1$, $X_\delta = 0.2$

6. RESULT DISCUSSION

The CEC2005 benchmark suite was used to evaluate the performance of the proposed Eight-GWO. The results of the proposed Eight-GWO were compared with those of Standard-GWO and PSO. The CEC2005 benchmark suite tests optimization algorithms by examining their ability to handle different levels of complexity.

Table 1: Parameters setting of Eight-GWO, GWO, and PSO

Parameter	*Eight-GWO*	*GWO*	*PSO*
P x D_p	30×30	30×30	30×30
Search space bounders	± 100	± 100	± 100
Max iterations	50	50	50
Vmax	-	-	6
WMax	-	-	0.9
WMin	-	-	0.2
c1, c2	-	-	2
S_{max}	10	-	-
D_{min} , D_{max}	[5,30]	-	-

Table 2 shows the result of the average 30 independent runs regrades to mean, median, best, and worst values in each one.

Table 2: Average of 30 independent runs by Eight-GWO, PSO, and GWO

Fun.	**Metrics**	*PSO*	*GWO*	*Eight-GWO*
F1	Mean	1054.138168	**184.706321**	232.2428271
	Median	847.6760134	138.5672376	216.2444551
	Best	285.1793502	66.69224899	94.02339504
	Worst	2459.224606	463.9014981	363.4273473
F2	Mean	51.10047101	5.988343826	**4.945009262**
	Median	46.79589896	5.569455104	4.434450945
	Best	29.59817916	4.469040403	3.420492624
	Worst	78.9189256	8.114951661	9.04773532
F3	Mean	13164.50615	11339.39963	**10064.27958**
	Median	12273.15713	11351.36794	9309.771325
	Best	6597.527419	5788.669636	4277.436937
	Worst	23389.03441	17195.43534	20064.13549
F4	Mean	20.553086	25.4436891	**18.75569258**
	Median	19.79493844	24.58235368	18.09558629
	Best	15.83235394	17.24911643	13.64540794
	Worst	25.14891108	35.40152933	23.71463314
F5	Mean	137904.0758	**19913.2638**	23428.90678
	Median	136214.8956	17605.63115	18423.51808
	Best	35258.55031	5058.029701	1894.988772
	Worst	328666.4809	42387.80596	59789.51065

F6	Mean	1078.779592	255.8387994	**188.0916763**
	Median	1008.836788	266.952305	190.5620866
	Best	526.1742237	57.3220391	60.66350796
	Worst	1866.281183	439.7603665	330.9353568
F7	Mean	33.28031906	0.245331459	**0.243996009**
	Median	35.0056403	0.260188641	0.211500281
	Best	14.35162676	0.101440715	0.083445477
	Worst	50.89267313	0.358117102	0.460555903
F8	Mean	-2440.61102	-3256.67229	**-4130.077215**
	Median	-2392.73987	-3009.07919	-4669.85251
	Best	-3224.99334	-5215.70129	-4982.721089
	Worst	-1790.96598	-2200.87246	-2385.448259
F9	Mean	320.4761376	190.0980729	**121.5579067**
	Median	318.2720187	155.026831	121.8967673
	Best	243.6403234	72.96438084	53.4693864
	Worst	379.8960568	342.0867852	176.2308427
F10	Mean	7.324847961	5.665358413	**4.660175021**
	Median	7.536412733	5.677880288	4.723437616
	Best	5.759002216	4.085778083	3.57263682
	Worst	9.428965138	7.18394192	6.385219579
F11	Mean	257.764338	3.242517518	**2.727990656**
	Median	259.6023529	3.233297821	2.626789908
	Best	212.281809	1.717345236	1.799620715
	Worst	297.1732647	5.091266082	4.080761408
F12	Mean	26.24830594	21.3353192	**14.23731089**
	Median	21.17695237	18.63840238	14.26129755
	Best	14.48796707	8.821202713	8.253926888
	Worst	56.1527603	47.49805702	23.09510908
F13	Mean	5092.614313	75.10162969	**48.79338415**
	Median	4119.966366	60.59183473	47.39745642
	Best	134.2536901	14.35875963	19.68603361
	Worst	15416.25837	248.7940789	81.37142203
F14	Mean	9.645956325	11.68133424	**6.456472292**
	Median	7.873993063	12.80422523	3.96825034
	Best	1.9920309	2.044988043	2.982360568
	Worst	21.98840767	21.07269182	11.7302187
F15	Mean	-1.03161154	-1.02963086	**-1.031619097**
	Median	-1.03162817	-1.03162092	-1.031620721
	Best	-1.03162843	-1.03162700	-1.031626104
	Worst	-1.03152558	-1.01173646	-1.031610781
F16	Mean	**-1.03161907**	-1.03121599	-1.031608881
	Median	-1.03162753	-1.03161855	-1.031611928
	Best	-1.03162829	-1.03162703	-1.031628422
	Worst	-1.03156682	-1.02767234	-1.031567004
F17	Mean	**0.397893625**	0.63666444	0.488859427
	Median	0.397888396	0.399532652	0.398304505
	Best	0.39788736	0.39792198	0.397985244
	Worst	0.397927773	2.705450285	1.288150039
F18	Mean	19.24253825	3.025969543	**3.018282041**
	Median	3.000004389	3.018194729	3.007742277
	Best	3.000000634	3.002066645	3.000383274
	Worst	84.41509301	3.084483204	3.105115309
F19	Mean	-3.82638509	**-3.84812119**	-3.845688033
	Median	-3.86277767	-3.86181295	-3.846451226
	Best	-3.86278205	-3.86252491	-3.861318238
	Worst	-3.51466078	-3.80508905	-3.82508617
F20	Mean	-2.83942740	-3.1338615	**-3.247233326**
	Median	-3.0775453	-3.17170667	-3.316123914

	Best	-3.32140393	-3.32105656	-3.31960787
	Worst	-1.70385981	-2.46152204	-3.112268989
F21	Mean	-4.48285803	-5.99711332	**-6.418504114**
	Median	-2.65941220	-6.06853195	-7.05067165
	Best	-10.1466929	-9.88278152	-10.01131663
	Worst	-2.62688415	-0.85651116	-2.557245454
F22	Mean	-4.77115027	-6.56404120	**-8.068106051**
	Median	-3.90001019	-7.43530393	-10.03578808
	Best	-10.3777947	-10.3244122	-10.34618127
	Worst	-1.83736142	-1.83081004	-2.752874841
F23	Mean	-4.39524934	-4.70971959	**-8.036502502**
	Median	-2.87016150	-2.58107176	-9.871776184
	Best	-10.5200407	-10.4378230	-10.44829117
	Worst	-1.67626435	-1.36894282	-2.390389738
F24	Mean	7.522340302	5.795988367	**5.056685453**
	Median	7.299833317	5.547303377	5.240551964
	Best	6.384820661	4.331429175	3.905496703
	Worst	10.04990265	7.738078747	5.95261499
F25	Mean	262.7598764	3.567901951	**3.014139304**
	Median	266.2726634	2.965835171	2.913141806
	Best	221.0655109	1.689364147	2.032516132
	Worst	291.927323	10.35999068	4.199497302

Unimodal functions require precise convergence to a single global optimum, which evaluates the precision and accuracy of an algorithm. Multimodal functions with multiple local optima test the algorithm's robustness and ability to avoid suboptimal solutions, a common risk for premature convergence. Hybrid functions often need to balance between exploration and exploitation across multiple "landscapes" of optimization, that is, if they should not flounder completely in any one place while foraging with the hope that somewhere, there may be something worth a piece of its effort. This means that the CEC2005 test suite offers an extremely severe and challenging environment for an algorithm's ability to produce fine solutions in many different kinds of situations consistently

The proposed Eight-Figure Grey Wolf Optimizer (Eight-GWO) can overcome the previously indicated problems. Exploring in a figure-eight pattern prevents the proposed algorithm from getting stuck at a local optimum. Moreover, it can help E-GWO jump out of holes in multimodal landscapes, thereby outperforming GWO and PSO in practice. It improved convergence and stability across multiple runs. That demonstrates its strength in maintaining consistent performance across several runs simultaneously. Figure 8 a illustrates the ability of the proposed Eight-GWO to escape local optima, demonstrating consistent results across multiple runs, as shown in the boxplot in Figure 8.b

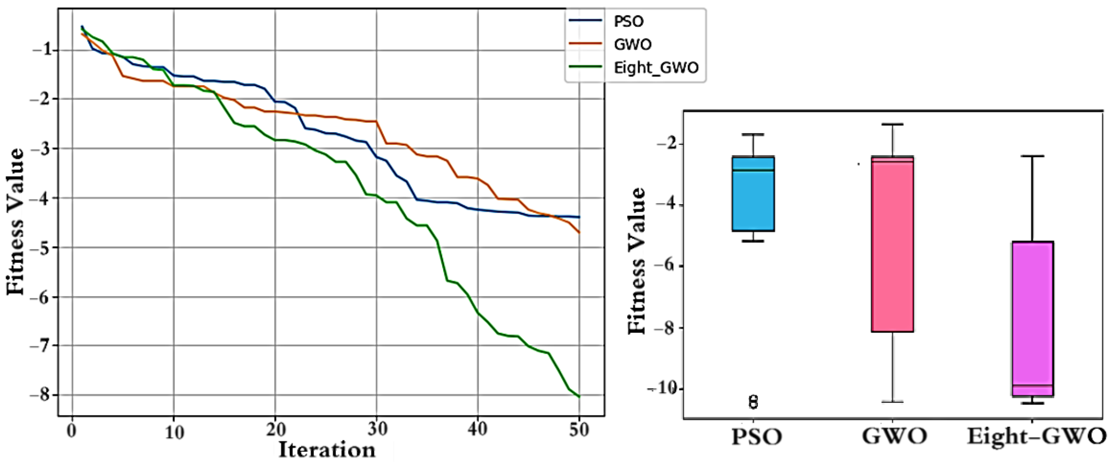

FIGURE 8: Performance of PSO, GWO, and Eight-GWO of F23 (Hybrid Composition) over 10 Individual Runs

Stagnation in local optima not only affects the efficiency of the optimization algorithm but also impacts its robustness. This occurs when the algorithm searches for extended periods without changing its state. It leads to a loss of time without meaningful progress. Therefore, the proposed Eight-Figure pattern framework forces the algorithm to switch its search strategy from the GWO to the Eight-Figure pattern. This shift also improves processing time since the proposed Eight-Figure pattern has a lower time complexity than the original GWO. The test performance of the comparative algorithms on the CEC2005 benchmark demonstrated that the proposed Eight-GWO required 23% less time than the original GWO and 44% less than the PSO.

7. CONCLUSION

This paper presents the eight-figure Gray-Wolf optimizer (eight-GWO) as a new version of GWO. It uses an adaptive Eight-Figure search pattern as an alternative searching tool. It enhanced the convergence, exploration, and reducing stagnation in local optima. The experimental results of the CEC2005 benchmark suite show that the proposed figure of eight GWO outperforms the standard GWO and PSO in terms of convergence speed, solution quality, and robustness. These improvements emphasize the effectiveness of an adaptive mechanism for balancing exploration and exploitation. Future work could include extending the eight-GWO to solve constrained optimization problems, multi-target tasks, and large-scale scenarios and integrating domain-specific knowledge to improve performance further.

REFERENCES

[1] A. H. Alsaeedi, D. Al-Shammary, S. M. Hadi, K. Ahmed, A. Ibaida, and N. AlKhazraji, "A proactive grey wolf optimization for improving bioinformatic systems with high dimensional data," *International Journal of Information Technology,* vol. 16, no. 8, pp. 4797-4814, 2024.

[2] T. Bikmukhametov and J. Jäschke, "Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models," *Computers & Chemical Engineering,* vol. 138, p. 106834, 2020.

[3] H. Zhang and Q. Peng, "PSO and K-means-based semantic segmentation toward agricultural products," *Future Generation Computer Systems,* vol. 126, pp. 82-87, 2022.

[4] R. R. Nuiaa, S. A. A. A. Alsaidi, B. K. Mohammed, A. H. Alsaeedi, Z. A. A. Alyasseri, S. Manickam, and M. A. Hussain, "Enhanced PSO Algorithm for Detecting DRDoS Attacks on LDAP Servers," *International Journal of Intelligent Engineering & Systems,* vol. 16, no. 5, 2023.

[5] R. R. Nuiaa, S. Manickam, A. H. Alsaeedi, and E. S. Alomari, "Enhancing the Performance of Detect DRDoS DNS Attacks Based on the Machine Learning and Proactive Feature Selection (PFS) Model," *IAENG International Journal of Computer Science,* vol. 49, no. 2, 2022.

[6] R. K. Deka, D. K. Bhattacharyya, and J. K. Kalita, "Active learning to detect DDoS attack using ranked features," *Computer Communications,* vol. 145, pp. 203-222, 2019.

[7] S. M. Ali, A. H. Alsaeedi, D. Al-Shammary, H. H. Alsaeedi, and H. W. Abid, "Efficient intelligent system for diagnosis pneumonia (SARSCOVID19) in X-ray images empowered with initial clustering," *Indones. J. Electr. Eng. Comput. Sci,* vol. 22, no. 1, pp. 241-251, 2021.

[8] A. H. Jabor and A. H. Ali, "Dual heuristic feature selection based on genetic algorithm and binary particle swarm optimization," *Journal of University of Babylon for Pure and Applied Sciences,* vol. 27, no. 1, pp. 171-183, 2019.

[9] L. Abualigah, "RETRACTED ARTICLE: Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications," *Neural Computing and Applications,* vol. 33, no. 7, pp. 2949-2972, 2021.

[10] T. Hamadneh, B. Batiha, G. M. Gharib, Z. Montazeri, F. Werner, G. Dhiman, M. Dehghani, R. K. Jawad, E. Aram, and I. K. Ibraheem, "Orangutan optimization algorithm: An innovative bio-inspired metaheuristic approach for solving engineering optimization problems," *Int. J. Intell. Eng. Syst,* vol. 18, no. 1, pp. 45-58, 2025.

[11] D. Freitas, L. G. Lopes, and F. Morgado-Dias, "Particle swarm optimisation: a historical review up to the current developments," *Entropy,* vol. 22, no. 3, p. 362, 2020.

[12] S. Du, W. Fan, and Y. Liu, "A novel multi-agent simulation based particle swarm optimization algorithm," *Plos one,* vol. 17, no. 10, p. e0275849, 2022.

[13] G. Negi, A. Kumar, S. Pant, and M. Ram, "GWO: a review and applications," *International Journal of System Assurance Engineering and Management,* vol. 12, pp. 1-8, 2021.

[14] H. Nozari and H. Abdi, "Greedy Man Optimization Algorithm (GMOA): A Novel Approach to Problem Solving with Resistant Parasites," *Journal of Industrial and Systems Engineering,* vol. 16, no. 3, pp. 106-117, 2024.

[15] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Applied Soft Computing,* vol. 60, pp. 115-134, 2017.

[16] F. A. Şenel, F. Gökçe, A. S. Yüksel, and T. Yiğit, "A novel hybrid PSO–GWO algorithm for optimization problems," *Engineering with Computers,* vol. 35, pp. 1359-1373, 2019.

[17] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, and A. Baroutaji, "The Role of Random Walk-Based Techniques in Enhancing Metaheuristic Optimization Algorithms—A Systematic and Comprehensive Review," *IEEE Access,* vol. 12, pp. 139573-139608, 2024, doi: 10.1109/ACCESS.2024.3466170.

[18] B. Çavdar, E. Şahin, and E. Sesli, "On the assessment of meta-heuristic algorithms for automatic voltage regulator system controller design: a standardization process," *Electrical Engineering,* vol. 106, no. 5, pp. 5801-5839, 2024/10/01 2024, doi: 10.1007/s00202-024-02314-x.

[19] Z. Ye, R. Huang, W. Zhou, M. Wang, T. Cai, Q. He, P. Zhang, and Y. Zhang, "Hybrid rice optimization algorithm inspired grey wolf optimizer for high-dimensional feature selection," *Scientific Reports,* vol. 14, no. 1, p. 30741, 2024/12/28 2024, doi: 10.1038/s41598-024-80648-z.

[20] H. Mohammed, Z. Abdul, and Z. Hamad, "Enhancement of GWO for solving numerical functions and engineering problems," *Neural Computing and Applications,* vol. 36, no. 7, pp. 3405-3413, 2024/03/01 2024, doi: 10.1007/s00521-023-09292-4.

[21] Z. Lyu, "State-of-the-Art Human-Computer-Interaction in Metaverse," *International Journal of Human–Computer Interaction,* pp. 1-19, 2023.

[22] D. Al‐Shammary, A. L. Albukhnefis, A. H. Alsaeedi, and M. Al‐Asfoor, "Extended particle swarm optimization for feature selection of high‐dimensional biomedical data," *Concurrency and Computation: Practice and Experience,* vol. 34, no. 10, p. e6776, 2022.

[23] M. Lozano and C. García-Martínez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report," *Computers & Operations Research,* vol. 37, no. 3, pp. 481-497, 2010.

[24] E. P. Krishna and A. Thangavelu, "Attack detection in IoT devices using hybrid metaheuristic lion optimization algorithm and firefly optimization algorithm," *International Journal of System Assurance Engineering and Management,* pp. 1-14, 2021.

[25] A. Kumar, S. Singh, and A. Kumar, "Grey wolf optimizer and other metaheuristic optimization techniques with image processing as their applications: a review," in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1136, no. 1: IOP Publishing, p. 012053.

[26] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software,* vol. 69, pp. 46-61, 2014.

[27] N. Hatta, A. M. Zain, R. Sallehuddin, Z. Shayfull, and Y. Yusoff, "Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014–2017)," *Artificial intelligence review,* vol. 52, pp. 2651-2683, 2019.