Review of Grey Wolf Optimizer

Preprint · May 2024

DOI: 10.13140/RG.2.2.14111.57763

1 author:

Dhruva Shaw
Lovely Professional University

Review of Grey Wolf Optimizer

Dhruva Shaw
Robotics and Automation
Lovely Professional University, India
dhruvashaw@ieee.org

***Abstract*—The Grey Wolf Optimizer (GWO), introduced in 2014, is a swarm intelligence optimization algorithm inspired by the social hierarchy and hunting behavior of grey wolves. This review delves into the core concepts of GWO, highlighting its strengths and limitations.**

Core Functionalities:

The GWO mimics the pack structure of wolves, with Alpha, Beta, Delta, and Omega wolves representing leadership roles. Alpha leads the hunt, Beta assists, Delta acts as a subordinate, and Omega follows. The algorithm simulates encircling prey, attacking, and searching by updating the positions of these virtual wolves based on their fitness values. This approach effectively balances exploration (finding promising areas) and exploitation (refining solutions within those areas).

Advantages:

GWO boasts several advantages over other swarm intelligence algorithms.

- **Simplicity: GWO requires minimal parameters for operation, making it easy to implement and use.**
- **Fast Convergence: The algorithm often reaches optimal solutions quickly, saving computational time.**
- **High Precision: GWO demonstrates a strong ability to find accurate solutions within the search space.**
- **Adaptability: GWO can be readily applied to various optimization problems due to its flexible design.**
- **Balance: The algorithm excels at maintaining a balance between exploration and exploitation, leading to robust solutions.**

Research Advancements: The core GWO framework has been actively extended and improved upon by researchers. These advancements include:

- **Hybrid approaches: Combining GWO with other algorithms like Particle Swarm Optimization (PSO) can leverage the strengths of both for enhanced performance.**
- **Modified versions: Variations like the Binary Grey Wolf Optimizer (BGWOPSO) address specific problem types, such as binary feature selection.**
- **Parallelization: Techniques for parallel computing can be applied to GWO, significantly reducing computation time for complex problems.**

Applications:

GWO's effectiveness has been demonstrated in diverse fields, including:

- **Machine Learning: GWO is used for feature selection, hyperparameter tuning, and model optimization.**
- **Engineering Design: The algorithm can optimize designs for efficiency, cost, and performance.**
- **Signal Processing: GWO finds applications in image and signal processing tasks.**

Future Directions:

Despite its success, GWO research continues to explore avenues for improvement. Potential areas include:

- **Dynamic parameter adaptation: Optimizing GWO parameters during the search process can enhance its performance in various scenarios.**
- **Handling complex constraints: Extending GWO's capabilities to deal with intricate problem constraints would broaden its applicability.**
- **Theoretical foundation: Further research into the theoretical underpinnings of GWO can lead to more robust performance guarantees.**

Conclusion:

The Grey Wolf Optimizer has established itself as a powerful and versatile optimization tool. This review has summarized its core principles, advantages, applications, and ongoing research directions. GWO's potential for tackling complex optimization problems across diverse fields makes it a valuable addition to the optimization toolbox.

***Index Terms*—Grey Wolf Optimizer (GWO), Swarm Intelligence Optimization, Bio-inspired Optimization, Metaheuristic Optimization, Exploration vs. Exploitation, Leadership Hierarchy, Encircling Prey, Attacking Prey, Searching Prey, Convergence Speed, Solution Precision, Adaptability, Hybrid Optimization, Binary Grey Wolf Optimizer (BGWOPSO), Parallelization, Machine Learning Applications, Engineering Design Applications, Signal Processing Applications, Dynamic Parameter Adaptation, Complex Constraint Handling, Theoretical Foundations, Feature Selection, Hyperparameter Tuning, Model Optimization, Efficiency Optimization, Cost Optimization, Performance Optimization, Image Processing, Signal Processing**

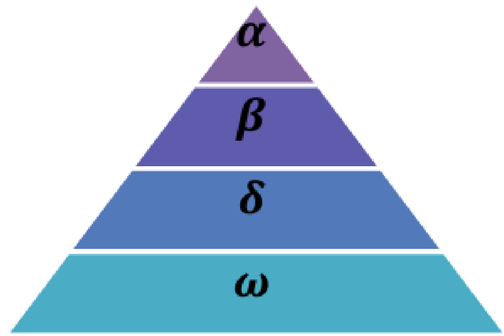

Fig. 1. Hierarchy of grey wolf (dominance decreases from top down) [1]

I. Unveiling the Power of the Pack: An Introduction to Grey Wolf Optimization

In the realm of optimization, where finding the best possible solution reigns supreme, nature continues to inspire innovative algorithms. One such algorithm, drawing inspiration from the social hierarchy and hunting prowess of grey wolves, is the

Grey Wolf Optimizer (GWO) [1]. Introduced in 2014 by Seyedali Mirjalili et al., GWO has emerged as a powerful tool for tackling complex optimization problems across diverse domains.

This introduction delves into the core principles of GWO, exploring its strengths and limitations while highlighting its versatility in various applications. We begin by setting the stage for optimization problems, where the goal is to identify the minimum or maximum value of a function within a defined search space. Traditional deterministic methods often struggle with intricate problems, particularly those with multiple potential solutions (local optima). Here's where metaheuristics come in.

Metaheuristics, a class of optimization algorithms, borrow inspiration from natural phenomena to guide the search towards optimal solutions. GWO falls within this category, mimicking the fascinating social structure and hunting behavior of grey wolves [1] [2]. These intelligent creatures exhibit a well-defined hierarchy, with alpha wolves leading the pack, followed by beta and delta wolves, and omega wolves holding the lowest rank. This hierarchy plays a crucial role in GWO's search process.

The hunting prowess of grey wolves is another key element mimicked by GWO. During a hunt, the pack works collaboratively. Alpha, beta, and delta wolves [1], representing the fittest solutions in GWO's terminology, guide the search for prey. Imagine these wolves as having knowledge of the prey's location. The remaining wolves (agents in GWO) strategically encircle and attack the prey, constantly updating their positions based on the information retrieved from the dominant wolves. Mathematically, GWO utilizes a set of equations to simulate these hunting behaviors. The fitness function, analogous to the prey's location, represents the objective of the search. Each wolf's position vector corresponds to a candidate solution in the search space [1] [2]. Through encircling and attacking equations, wolves adjust their positions, moving closer to the fittest solutions identified by the alpha, beta, and delta wolves. A control parameter, 'a', plays a vital role; it decreases over iterations, guiding the search from exploration (finding promising areas) to exploitation (refining the solution around the best found so far).

The simplicity of GWO is one of its key strengths. With a clear mathematical framework and fewer parameters compared to some algorithms, GWO is relatively easy to understand and implement. This translates to faster development cycles and easier integration into existing optimization frameworks. Additionally, GWO exhibits remarkable efficiency, particularly well-suited for continuous optimization problems. Its ability to handle complex search spaces with multiple local optima further underscores its robustness.

However, like any optimization algorithm, GWO also has limitations. Premature convergence, where the search gets stuck in local optima if not carefully tuned, is a potential concern. GWO might struggle with highly discontinuous or multimodal search spaces (where the objective function has multiple peaks and valleys). While having fewer parameters is an advantage,

Fig. 2. Hunting behaviour of grey wolves: (A) chasing, approaching, and tracking prey (B-D) pursuiting, harassing, and encircling (E) stationary situation and attack [1]

finding optimal parameter values can be problem-specific and require experimentation. Despite these limitations, GWO's strengths make it a valuable tool across diverse application domains. In machine learning, GWO can be employed for hyperparameter tuning, optimizing the settings of algorithms like Support Vector Machines or neural networks, to achieve superior performance.

Engineering design benefits from GWO's ability to optimize parameters in control systems, power system operation, or structural design, leading to more efficient and robust systems. The field of image processing harnesses GWO for tasks like image segmentation, feature extraction, or filter design, facilitating the development of advanced image analysis tools. The versatility of GWO extends beyond these examples, with applications in finance, scheduling, robotics, and data mining, to name a few. By harnessing the collective intelligence of the wolf pack, GWO offers a powerful and adaptable approach to tackling complex optimization challenges. As research in this field continues, we can expect further advancements in GWO, potentially through hybridization with other algorithms or tailored adaptations for specific problem types. As we explore the optimization landscape, the Grey Wolf Optimizer stands as a testament to the enduring power of nature to inspire innovative solutions.

II. Background: Optimization Problems and Metaheuristics

Before diving into the specifics of Grey Wolf Optimization (GWO), let's establish a foundation in optimization problems and metaheuristics. Understanding these concepts will provide a clearer context for GWO's functionality and its place within the optimization landscape.

A. 1. Optimization Problems: Finding the Best

Optimization problems are ubiquitous across various scientific and engineering disciplines. In essence, they involve finding the minimum or maximum value of a function within a defined search space. Imagine you're scaling a mountain; the goal is to reach the highest point (maximum) or the lowest

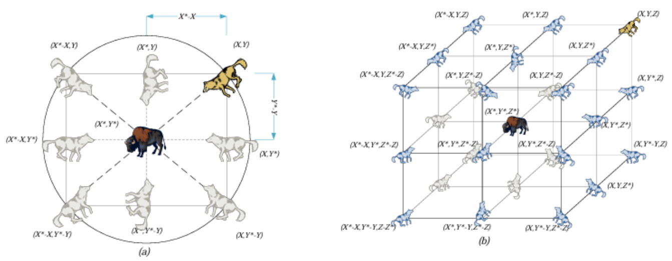

Fig. 3. 2D and 3D position vectors and their possible next locations [1]

valley (minimum) while navigating the complex terrain (search space). Traditional deterministic methods, such as calculus-based approaches, often struggle with intricate optimization problems. These challenges can arise due to:

- **Non-linearity**: The relationship between variables and the objective function might not be a smooth curve, making traditional methods less effective.
- **Multiple Local Optima**: The search space might contain multiple peaks or valleys (local optima) besides the global optimum (the absolute highest or lowest point). Deterministic methods can get trapped in these local optima, failing to reach the best solution.

B. 2. Metaheuristics

Borrowing from Nature for Smarter Search This is where metaheuristics step in. Metaheuristics are a class of optimization algorithms that draw inspiration from natural phenomena to guide the search towards optimal solutions. They don't rely on intricate calculations about the function itself, but rather utilize high-level strategies inspired by natural processes. Here are some key characteristics of metaheuristics:

- **Iterative Approach**: They work in an iterative manner, gradually refining the search based on evaluations of candidate solutions.
- **Stochastic Elements**: They often incorporate a degree of randomness to explore the search space effectively and avoid getting stuck in local optima.
- **Balance between Exploration and Exploitation** [3]: Metaheuristics strive for a balance between exploration (finding promising areas of the search space [1] [3]) and exploitation (refining the search around the best solutions found so far).

C. 3. GWO: Inspired by the Wolf Pack

Grey Wolf Optimization (GWO) is a specific type of metaheuristic that mimics the social hierarchy and hunting behavior of grey wolves [1] [3]. These intelligent creatures exhibit a well-defined social structure, with alpha, beta, and delta wolves leading the pack, and omega wolves holding the lowest rank. GWO leverages this hierarchy to guide the search process, as we'll explore further in the next section [2].

III. Grey Wolf Optimization Algorithm

Grey Wolf Optimization (GWO) is a powerful metaheuristic algorithm inspired by the social hierarchy and hunting behavior of grey wolves [1] [3]. It tackles complex optimization problems by mimicking these natural processes, offering an efficient and robust approach. This section delves deeper into the core principles of GWO, unpacking its key formulas and how they translate to the optimization process [1] [2] [3].

A. The Wolf Pack Hierarchy: Guiding the Search

GWO establishes a virtual wolf pack, where each wolf represents a candidate solution in the search space. The pack adheres to a well-defined hierarchy:

- **Alpha** (α): Represents the fittest solution (with the best fitness value) found so far [1].
- **Beta** (β): The second-best solution in the pack [1].
- **Delta** (δ): The third-best solution identified during the search [1].
- **Omega** (ω): All remaining wolves (candidate solutions) constitute the omega group [1].

These rankings dynamically change as the search progresses. The alpha, beta, and delta wolves guide the search for the optimal solution, influencing the movements of the omega wolves [1] [2] [3].

B. The Encircling Mechanism: Closing in on the Prey

Imagine the wolves tracking and encircling their prey (the optimal solution). GWO utilizes the following formula to simulate this encircling behaviour:

- $D =\mid C \times X_{p(t)} - X(t) \mid$ [1]

Here:

-
- represents the distance vector between a specific wolf (X(t)) and the prey ($X_{p(t)}$) at a particular iteration (t) [1].
- **C** is a coefficient vector randomly generated within the range [0, 2] [1].
- ($X_{p(t)}$) denotes the position vector of the prey (current best solution: alpha, beta, or delta) [1].
- **X(t)** represents the position vector of a specific wolf (candidate solution) at iteration t [1].

By calculating the distance vector (D), GWO determines how close each wolf is to the prey.

C. The Hunting Equations: Attacking the Prey and Updating Positions

The hunting phase involves wolves strategically attacking the prey and updating their positions based on the information obtained from alpha, beta, and delta. Two key formulas govern this process:

- $X(t+1) = X_{p(t)} - A \times D(t)$

Here:

- **X(t+1)** represents the updated position vector of a specific wolf in the next iteration (t+1).

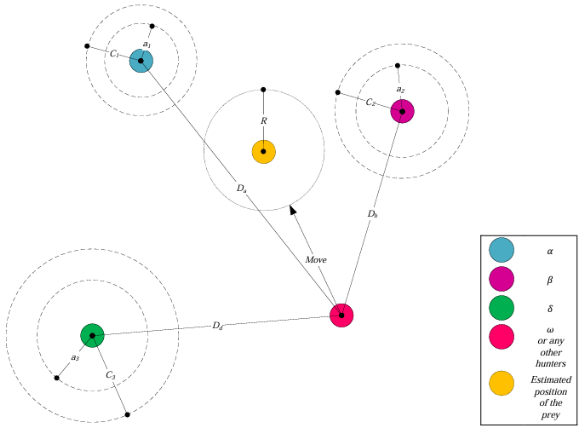

Fig. 4. Position updading in GWO [1]

- **A** is a control parameter that linearly decreases from 2 to 0 over iterations. It mimics the hunting process, where wolves initially focus on exploration (higher A values) and gradually shift towards exploitation (lower A values) as they converge on the prey.
- $D(t) = C1 \times X_{alpha(t)} - X(t) + C2 \times X_{beta(t)} - X(t) + C3 \times X_{delta(t)} - X(t)$ This formula calculates the direction in which a specific wolf (X(t)) needs to move to get closer to the prey.
- **C1**, **C2**, **C3** are random coefficients generated within the range [-1, 1]. They introduce stochasticity, preventing the wolves from converging too quickly and potentially getting stuck in local optima [1] [2] [3].
- $X_{alpha(t)}, X_{beta(t)}, X_{delta(t)}$ represents the position vectors of alpha, beta, and delta wolves at the current iteration (t) [1] [2] [3].

These formulas essentially calculate the updated position of each omega wolf based on its distance to the prey and the positions of the alpha, beta, and delta wolves. By iteratively applying these equations, the entire pack refines its search, collectively converging towards the optimal solution.

D. The Omega Wolf Update: Balancing Exploration and Exploitation

It's important to note that the alpha, beta, and delta wolves (the fittest solutions) don't directly update their positions using these formulas. They guide the search by influencing the movements of the omega wolves. This maintains a balance between exploration (omega wolves venturing into new areas) and exploitation (the pack converging on promising areas identified by the leading wolves) [1] [2] [4].

IV. Strengths and Weaknesses of GWO

A. Strengths of Grey Wolf Optimization (GWO)

GWO boasts several advantages that make it a valuable optimization tool:

- **Simplicity**: Compared to some algorithms, GWO's mathematical framework is relatively straightforward. With fewer parameters to tune, it's easier to understand and implement, leading to faster development cycles and simpler integration into existing optimization frameworks.
- **Efficiency**: GWO excels at finding good solutions quickly, particularly for continuous optimization problems. Its efficient search process converges well on the optimal solution without getting bogged down in unnecessary iterations.
- **Robustness**: GWO demonstrates remarkable resilience in handling complex search spaces with multiple local optima. Unlike deterministic methods that can get trapped in these local minima/maxima, GWO's stochastic elements and exploration-exploitation balance allow it to navigate these challenges effectively.
- **Fewer Parameters**: Having fewer parameters compared to some other metaheuristics translates to less time spent on parameter tuning. However, it's important to note that finding optimal parameter values can still be problem-specific and require some experimentation.

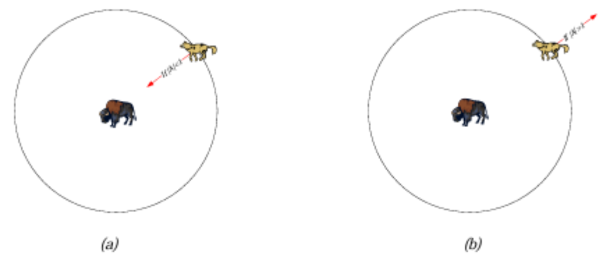

Fig. 5. Attacking prey versus searching for prey [1]

Initialize the grey wolf population X_i (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
X_α=the best search agent
X_β=the second best search agent
X_δ=the third best search agent
***while** (t < Max number of iterations)*
    ***for** each search agent*
        Update the position of the current search agent by equation (3.7)
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    t=t+1
end while
return X_α

Fig. 6. Pseudo code of the GWO algorithm [1]

B. Weaknesses of Grey Wolf Optimization (GWO)

While powerful, GWO also has limitations to consider:

- **Premature Convergence**: If not carefully tuned, GWO can exhibit premature convergence, where the search gets stuck in a local optimum and fails to reach the global optimum [4] [5] [3] [1] [2]. This highlights the importance of proper parameter selection and potentially incorporating mechanisms to prevent stagnation.

- **Limited Exploration**: For highly discontinuous or multi-modal search spaces (with multiple peaks and valleys in the objective function), GWO's exploration capabilities might be limited. The stochastic elements might not be sufficient to thoroughly explore all promising areas, potentially leading to suboptimal solutions.
- **Parameter Tuning**: Despite having fewer parameters, finding the optimal values for these parameters can be problem specific. This requires some experimentation and may not always be straightforward, especially for complex optimization problems.

Overall, GWO offers a compelling balance between simplicity, efficiency, and robustness. While it has limitations, its strengths make it a valuable tool for a wide range of optimization tasks. By understanding both its strengths and weaknesses, you can effectively apply GWO to various problems and potentially explore avenues for further improvement through parameter tuning or hybridization with other algorithms.

V. Application sof GWO

Grey Wolf Optimization (GWO), inspired by the collaborative hunting prowess of grey wolves[1], has transcended its biological roots to become a versatile tool across diverse application domains. Its ability to find optimal solutions in complex scenarios makes it an asset in various fields. Let's delve into some of the prominent applications of GWO:

A. Machine Learning: Optimizing Algorithms for Peak Performance

- **Hyperparameter Tuning**: GWO excels at optimizing the hyperparameters of machine learning algorithms like Support Vector Machines (SVMs) or neural networks [4]. These hyperparameters significantly influence the performance of the algorithm, and GWO can efficiently find the optimal settings that lead to superior accuracy, generalization, and efficiency [5].
- **Feature Selection**: Feature selection, the process of choosing the most relevant features for a machine learning model [2], can benefit from GWO. By identifying the most informative features, GWO helps build more efficient and accurate models with reduced training time and complexity.

B. Engineering Design: Optimizing Systems for Efficiency and Performance

- **Control System Design**: GWO can optimize the parameters of control systems, ensuring stability, desired response characteristics, and efficient operation. This is crucial in various engineering applications, from automated manufacturing systems to robotics and process control.
- **Power System Operation**: Optimizing power system operation involves balancing supply and demand while minimizing costs. GWO can effectively handle the complexities of power systems, including variable renewable energy sources and fluctuating loads, to ensure reliable and efficient power delivery.
- **Structural Design**: In structural engineering, GWO can optimize the design of structures for strength, stability, and material usage. By finding the optimal configuration and material distribution, GWO helps create strong, lightweight, and cost-effective structures.

Function	Dim	Range	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[-100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-100,100]	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	[-1.28,1.28]	0

Fig. 7. Unimodal benchmark functions [1]

C. Image Processing: Enhancing Image Analysis Tools

- **Image Segmentation**: GWO can be employed for image segmentation, which involves dividing an image into meaningful regions. This is crucial for object recognition, medical image analysis, and other applications. GWO can find optimal segmentation parameters, leading to more accurate and robust segmentation results.
- **Feature Extraction**: Feature extraction is the process of identifying key characteristics from images. GWO can optimize feature extraction algorithms, ensuring that the extracted features are relevant and informative for subsequent analysis tasks.
- **Filter Design**: Image filters are used for various purposes, such as noise reduction and image enhancement. GWO can optimize the design of these filters, leading to superior image quality and clearer visualizations.

D. Beyond these Examples: A Broader Impact

The reach of GWO extends far beyond the listed examples. Its versatility makes it applicable in various other domains, including:

- **Finance**: Optimizing investment portfolios, risk management, and financial forecasting.
- **Scheduling**: Creating efficient schedules for resource allocation, production planning, and project management.
- **Robotics**: Optimizing robot motion planning, pathfinding algorithms, and control parameters.
- **Data Mining**: Identifying patterns and extracting knowledge from large datasets.

Function	Dim	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i \rvert})$	30	[-500,500]	-418.9829×5
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$F_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	30	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$F_{12}(x) =$ $\frac{\pi}{n}\{10sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$F_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	30	[-50,50]	0
$F_{14}(x) = -\sum_{i=1}^{n} \sin(x_i) . \left(\sin\left(\frac{i x_i^2}{\pi}\right)\right)^{2m}, \quad m = 10$	30	[0,π]	-4.687
$F_{15}(x) = \left[e^{-\sum_{i=1}^{n}(x_i/\beta)^{2m}} - 2e^{-\sum_{i=1}^{n} x_i^2}\right] . \prod_{i=1}^{n} \cos^2 x_i, \quad m = 5$	30	[-20,20]	-1
$F_{16}(x) = \{[\sum_{i=1}^{n} \sin^2(x_i)] - exp(-\sum_{i=1}^{n} x_i^2)\} . exp[-\sum_{i=1}^{n} \sin^2 \sqrt{\lvert x_i \rvert}]$	30	[-10,10]	-1

Fig. 8. Multimodal benchmark functions [1]

Function	Dim	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[-5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	2	[-5,5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[1,3]	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0,1]	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

Fig. 9. Fixed-dimenstion multimodal benchmark functions [1]

VI. A Comparative Analysis: Grey Wolf Optimization (GWO) vs. Other Algorithms

Grey Wolf Optimization (GWO) has emerged as a prominent player in the realm of optimization algorithms. However, it's valuable to compare GWO with other popular algorithms to understand its relative strengths and weaknesses in different scenarios. Here's a breakdown of GWO's performance against some common competitors:

Function	Dim	Range	f_{min}
F_{24}(CF1): $f_1, f_2, f_3, \dots, f_{10}$ = Sphere Function $[б_1, б_2, б_3, \dots, б_{10}] = [1,1,1,..,1]$ $[\lambda_1, \lambda_2, \lambda_3 \dots, \lambda_{10}] = [5/100, 5/100, 5/100, .., 5/100]$	10	[-5,5]	0
F_{25}(CF2): $f_1, f_2, f_3, \dots, f_{10}$ = *Griewank's Function* $[б_1, б_2, б_3, \dots, б_{10}] = [1,1,1,..,1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/100, 5/100, 5/100, .., 5/100]$	10	[-5,5]	0
F_{26}(CF3): $f_1, f_2, f_3, \dots, f_{10}$ = *Griewank's Function* $[б_1, б_2, б_3, \dots, б_{10}] = [1,1,1,..,1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [1,1,1,..,1]$	10	[-5,5]	0
F_{27}(CF4): f_1, f_2 = Ackley'sFunction f_3, f_4 = Rastrigin's Function f_5, f_6 = Weierstrass Function f_7, f_8 = Griewank's Function f_9, f_{10} = Sphere Function $[б_1, б_2, б_3, \dots, б_{10}] = [1,1,1,..,1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10	[-5,5]	0
F_{28}(CF5): f_1, f_2 = Rastrigin's Function f_3, f_4 = Weierstrass Function f_5, f_6 = Griewank's Function f_7, f_8 = Ackley'sFunction f_9, f_{10} = Sphere Function $[б_1, б_2, б_3, \dots, б_{10}] = [1,1,1,..,1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	[-5,5]	0
f_{29}(CF6): f_1, f_2 = Rastrigin's Function f_3, f_4 = Weierstrass Function f_5, f_6 = Griewank's Function f_7, f_8 = Ackley'sFunction f_9, f_{10} = Sphere Function $[б_1, б_2, б_3, \dots, б_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	10	[-5,5]	0

Fig. 10. Composite benchmark functions [1]

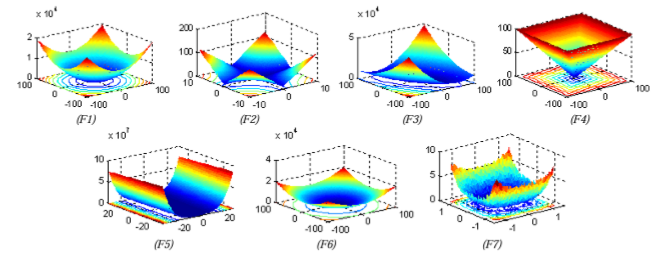

Fig. 11. 2-D versions of unimodal benchmark functions [1]

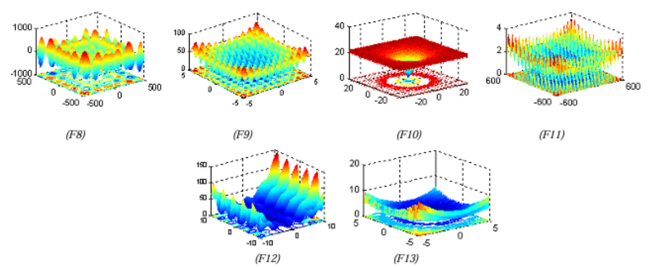

Fig. 12. 2-D versions of multimodal benchmark functions [1]

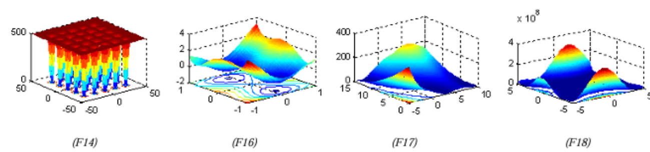

Fig. 13. 2-D versions of composite benchmark functions [1]

F	GWO		PSO		GSA		DE		FEP	
	ave	*std*	*ave*	*std*	*ave*	*std*	*ave*	*std*	*ave*	*std*
F1	6.59E-28	6.34E-05	0.000136	0.000202	2.53E-16	9.67E-17	8.2E-14	5.9E-14	0.00057	0.00013
F2	7.18E-17	0.029014	0.042144	0.045421	0.055655	0.194074	1.5E-09	9.9E-10	0.0081	0.00077
F3	3.29E-06	79.14958	70.12562	22.11924	896.5347	318.9559	6.8E-11	7.4E-11	0.016	0.014
F4	5.61E-07	1.315088	1.086481	0.317039	7.35487	1.741452	0	0	0.3	0.5
F5	26.81258	69.90499	96.71832	60.11559	67.54309	62.22534	0	0	5.06	5.87
F6	0.816579	0.000126	0.000102	8.28E-05	2.5E-16	1.74E-16	0	0	0	0
F7	0.002213	0.100286	0.122854	0.044957	0.089441	0.04339	0.00463	0.0012	0.1415	0.3522

Fig. 14. Results of unimodal benchmark functions [1]

F	GWO		PSO		GSA		DE		FEP	
	ave	std	ave	std	ave	std	ave	std	ave	std
F8	-6123.1	-4087.44	-4841.29	1152.814	-2821.07	493.0375	-11080.1	574.7	-12554.5	52.6
F9	0.310521	47.35612	46.70423	11.62938	25.96841	7.470068	69.2	38.8	0.046	0.012
F10	1.06E-13	0.077835	0.276015	0.50901	0.062087	0.23628	9.7E-08	4.2E-08	0.018	0.0021
F11	0.004485	0.006659	0.009215	0.007724	27.70154	5.040343	0	0	0.016	0.022
F12	0.053438	0.020734	0.006917	0.026301	1.799617	0.95114	7.9E-15	8E-15	9.2E-06	3.6E-06
F13	0.654464	0.004474	0.006675	0.008907	8.899084	7.126241	5.1E-14	4.8E-14	0.00016	0.000073

Fig. 15. Results of multimodal benchmark functions [1]

F	GWO		PSO		GSA		DE		FEP	
	ave	std	ave	std	ave	std	ave	std	ave	std
F14	4.042493	4.252799	3.627168	2.560828	5.859838	3.831299	0.998004	3.3E-16	1.22	0.56
F15	0.000337	0.000625	0.000577	0.000222	0.003673	0.001647	4.5E-14	0.00033	0.0005	0.00032
F16	-1.03163	-1.03163	-1.03163	6.25E-16	-1.03163	4.88E-16	-1.03163	3.1E-13	-1.03	4.9E-07
F17	0.397889	0.397887	0.397887	0	0.397887	0	0.397887	9.9E-09	0.398	1.5E-07
F18	3.000028	3	3	1.33E-15	3	4.17E-15	3	2E-15	3.02	0.11
F19	-3.86263	-3.86278	-3.86278	2.58E-15	-3.86278	2.29E-15	N/A	N/A	-3.86	0.000014
F20	-3.28654	-3.25056	-3.26634	0.060516	-3.31778	0.023081	N/A	N/A	-3.27	0.059
F21	-10.1514	-9.14015	-6.8651	3.019644	-5.95512	3.737079	-10.1532	0.0000025	-5.52	1.59
F22	-10.4015	-8.58441	-8.45653	3.087094	-9.68447	2.014088	-10.4029	3.9E-07	-5.53	2.12
F23	-10.5343	-8.55899	-9.95291	1.782786	-10.5364	2.6E-15	-10.5364	1.9E-07	-6.57	3.14

Fig. 16. Results of fixed-dimension multimodal benchmark functions [1]

F	GWO		PSO		GSA		DE		CMA-ES	
	ave	*std*	*ave*	*std*	*ave*	*std*	*ave*	*std*	*ave*	*std*
F24	43.83544	69.86146	100	81.65	6.63E-17	2.78E-17	6.75E-02	1.11E-01	100	188.56
F25	91.80086	95.5518	155.91	13.176	200.6202	67.72087	28.759	8.6277	161.99	151
F26	61.43776	68.68816	172.03	32.769	180	91.89366	144.41	19.401	214.06	74.181
F27	123.1235	163.9937	314.3	20.066	170	82.32726	324.86	14.784	616.4	671.92
F28	102.1429	81.25536	83.45	101.11	200	47.14045	10.789	2.604	358.3	168.26
F29	43.14261	84.48573	861.42	125.81	142.0906	88.87141	490.94	39.461	900.26	8.32E-02

Fig. 17. Results of composite benchmark functions [1]

A. Particle Swarm Optimization (PSO)

1) **Performance**:
    - GWO often converges faster than PSO, especially for continuous optimization problems.
    - PSO might struggle with complex search spaces with multiple optima [4].
2) **Complexity**:
    - Both GWO and PSO have relatively simple implementations with few parameters.
3) **Strengths and Weaknesses**:
    - GWO excels in exploration and exploitation balance, while PSO might get stuck in local optima.

B. Genetic Algorithm (GA)

1) **Performance**:
    - GWO can be faster than GA, particularly in problems with complex search spaces.
    - GA might perform better in highly discontinuous or multimodal problems due to its crossover operator [5].
2) **Complexity**:
    - GA has a more complex implementation with operators like selection, crossover, and mutation, requiring more tuning effort [5].
3) **Strengths and Weaknesses**:
    - GWO offers efficient search and fewer parameters, while GA provides better diversity with its crossover operator.

C. Differential Evolution (DE)

1) **Performance**:
    - GWO and DE exhibit comparable convergence speeds for various optimization problems.
    - DE might be slightly more robust for noisy or ill-conditioned problems [5].
2) **Complexity**:
    - DE has a slightly higher parameter tuning complexity compared to GWO.
3) **Strengths and Weaknesses**:
    - GWO offers a clear social hierarchy for search guidance, while DE relies on mutation for exploration [5].

D. Whale Optimization Algorithm (WOA)

1) **Performance**:
    - GWO and WOA demonstrate similar convergence speeds in many cases.
    - WOA might excel in specific problems due to its exploration-exploitation strategy inspired by humpback whale hunting [5].
2) **Complexity**:
    - Both algorithms have comparable implementation complexity.
3) **Strengths and Weaknesses**:
    - GWO focuses on social hierarchy for search, while WOA mimics specific whale behaviors for exploration and exploitation [5].

E. Key Considerations When Choosing an Algorithm

- **Problem Type**: The specific characteristics of your optimization problem (continuous, discrete, multimodal) can influence the choice of algorithm [1].
- **Performance**: Convergence speed, solution quality, and robustness to local optima should be considered [6].
- **Complexity**: Ease of implementation and parameter tuning can impact development time [6].

VII. Grey Wolf Optimizer for Electromagnetic Field (EMF) Applications

The Grey Wolf Optimizer (GWO) presents a powerful and versatile approach for tackling optimization problems in the field of electromagnetics (EMF) [7]. This section delves into the theoretical underpinnings of applying GWO to EMF challenges [8], explores its implementation strategies, and highlights the advantages this methodology offers.

A. Electromagnetic Field Optimization Problems

EMF problems often involve optimizing design parameters to achieve specific performance goals [9]. These goals can encompass various aspects of electromagnetic wave behavior, including:

- **Radiation Characteristics**: Optimizing antenna directivity, gain, radiation pattern, and frequency response for efficient transmission or reception [10].
- **Wave Propagation Control**: Manipulating material properties to achieve desired wave propagation behavior, such as focusing, filtering, or shielding [11].
- **Energy Transfer**: Optimizing coupling coefficients between antennas or maximizing power transfer efficiency in wireless power transmission systems [12].

The complexity of these problems often necessitates the use of computational tools. Traditional optimization techniques can be time-consuming or struggle to escape local minima in the search space [12]. GWO offers a compelling alternative due to its strengths in [13]:

- **Balancing Exploration and Exploitation**: GWO effectively balances exploration (finding promising regions

in the search space) and exploitation (refining solutions within those regions) [14]. This leads to a robust search process that can identify optimal solutions across diverse EMF problems. [15]
- **Minimal Parameter Tuning**: GWO requires minimal parameter configuration compared to other optimization algorithms, simplifying its implementation for researchers and engineers [14].

B. GWO Implementation for EMF Applications

The application of GWO to EMF problems involves several key steps:

1) **Problem Encoding**
   The first step involves encoding the design parameters of the EMF system into a format suitable for GWO. This typically involves representing each candidate solution (a potential antenna design or material configuration) as a position vector within a multi-dimensional search space. The specific dimensions of this vector depend on the problem at hand. [16] For instance, in antenna design optimization, the vector might represent antenna dimensions (length, width, etc.), material properties (permittivity, permeability), and positioning of feed points. Each element in the vector corresponds to a design parameter that GWO will optimize [5].

2) **Fitness Function Definition**
   An objective function, also known as a fitness function, is defined to evaluate the performance of each candidate solution (wolf position) in the GWO pack. This function quantifies the degree to which a particular design configuration meets the desired goals [4]. Depending on the specific EMF application, the fitness function might consider various factors [5]: newline

3) **Antenna Performance Metrics**
   For antenna design problems, the fitness function might evaluate parameters like antenna gain, directivity, return loss, bandwidth, or a combination of these, depending on the desired functionality [16].
   - **Wave Propagation Characteristics**: In problems involving wave control, the fitness function could assess factors like field intensity at specific locations, reflection or transmission coefficients of materials, or the degree to which the wave interacts with a desired target.
   - **Energy Transfer Efficiency**: For wireless power transmission applications, the fitness function might quantify the amount of power transferred from the transmitter to the receiver, taking into account factors like coupling efficiency and power losses.

4) **GWO Iteration Process**
   The core GWO algorithm is then executed. This involves iteratively updating the positions of the virtual wolves (candidate solutions) based on the fitness function [1]. The algorithm mimics the social hierarchy and hunting behavior of grey wolves, where:
   - **Alpha Wolf**: The wolf with the best fitness value (highest performance according to the objective function) is designated as the Alpha wolf [10].
   - **Beta and Delta Wolves**: The wolves with the second and third best fitness values become the Beta and Delta wolves [17], respectively. These wolves guide the search process towards promising areas [18].
   - **Omega Wolves**: The remaining wolves represent Omega wolves, and their positions are updated based on the positions of Alpha, Beta, and Delta [14]. This allows for exploration of the search space and prevents premature convergence on local optima.

   Through a series of mathematical calculations, each wolf updates its position in the search space, effectively iterating towards improved solutions. The iteration process continues until a predefined stopping criterion is met. This criterion could be reaching a maximum number of iterations, achieving a desired level of fitness, or a combination of both [5].

C. Advantages of GWO for EMF Applications

GWO offers several distinct advantages over traditional optimization techniques when applied to EMF problems:
- **Fast Convergence**: As mentioned earlier, GWO's ability to quickly converge to optimal solutions makes it ideal for computationally expensive electromagnetic simulations [4]. This can significantly reduce design time and computational resources.
- **High Precision**: The algorithm demonstrates a strong capability of finding highly accurate solutions within the search space. This translates to well-designed electromagnetic components with optimal performance characteristics [19].

D. Adaptability

GWO's framework excels in its ability to adapt to various EMF problems. This stems from two key aspects:

- **Flexible Fitness Function**: The core of GWO remains the same – mimicking wolf pack behavior for optimization. However, the **fitness function** can be readily customized to address specific problems in EMF. This function defines what constitutes a "good" solution for the specific design challenge [14].
- **Minimal Parameter Tuning**: Unlike some optimization algorithms, GWO requires minimal parameter configuration. This makes it user-friendly and reduces the need for extensive fine-tuning for different EMF applications [20].

E. Additional Advantages

- **Robustness**: GWO exhibits a strong ability to avoid getting trapped in local minima (suboptimal solutions) in the search space. This is due to the exploration-exploitation balance inherent in the algorithm [20].
- **Parallelization Potential**: GWO can be readily adapted for parallel computing environments. This allows for significant speedup in solving complex EMF problems that require intensive computations [21].

F. Comparison with Traditional Methods

Traditional optimization techniques used in electromagnetics often involve gradient-based methods or exhaustive search approaches. While these methods can be effective, they may suffer from drawbacks:

- **Slow Convergence**: Gradient-based methods can converge slowly, especially for complex problems with many design variables.
- **Local Minima**: These methods can get trapped in local minima, leading to suboptimal solutions [10].
- **High Computational Cost**: Exhaustive search approaches can be computationally expensive, especially for problems with large design spaces [17].

GWO offers a compelling alternative by addressing these limitations. Its fast convergence, ability to escape local minima, and lower computational cost make it a valuable tool for EMF design optimization [14].

G. Future Directions

Research on GWO for EMF applications is an ongoing endeavor. Potential areas for future exploration include:

- **Hybrid Approaches**: Combining GWO with other optimization techniques like Particle Swarm Optimization (PSO) could leverage the strengths of both for even better performance [5].
- **Problem-Specific GWO Variants**: Developing specialized GWO variants tailored to address specific classes of EMF problems could further enhance its effectiveness [4].
- **Theoretical Analysis**: Further research into the theoretical underpinnings of GWO's performance in the context of EMF applications can provide deeper insights and guide future algorithm development [19].

By capitalizing on GWO's strengths and exploring these potential advancements, researchers and engineers can unlock the full potential of this optimization technique for tackling diverse challenges in the field of electromagnetics [19].

VIII. Conclusion: GWO - A Powerful Tool Inspired by Nature

The Grey Wolf Optimizer (GWO) has emerged as a powerful and versatile tool for tackling complex optimization problems across diverse scientific and engineering fields. Its core

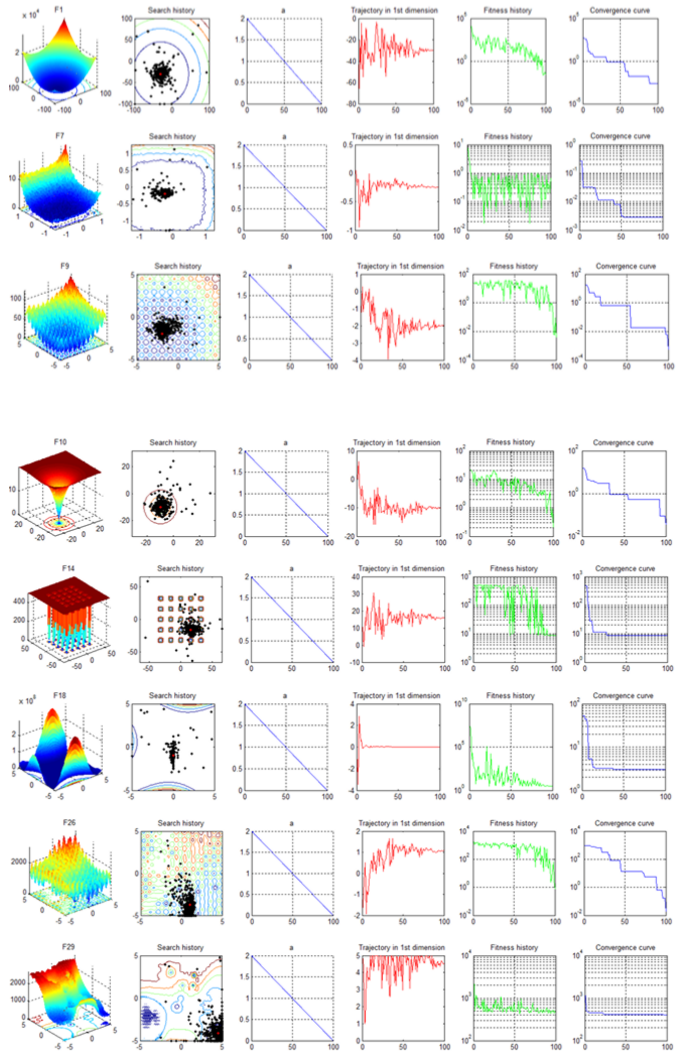

Fig. 18. Search history and trajectory of the first particle in the first dimension [1]

strength lies in mimicking the social hierarchy and hunting behavior of grey wolves, translating these natural phenomena into a robust mathematical framework for optimization. This conclusion section delves into the key characteristics that make GWO a compelling choice for researchers and engineers.

1) **Clear Mathematical Framework**: Unlike some optimization algorithms with intricate mathematical underpinnings, GWO boasts a relatively straightforward mathematical structure. This clarity allows for easier implementation and adaptation to various problem domains. Researchers can readily understand the underlying principles of GWO, facilitating modifications and customizations for specific optimization challenges.
2) **Efficient Search Process**: GWO excels in its ability to efficiently navigate the search space towards optimal solutions. The algorithm effectively balances exploration (finding promising areas) and exploitation (refining solutions within those areas). This balance is achieved

through the concept of Alpha, Beta, and Delta wolves guiding the search, while Omega wolves explore the broader space. This dynamic interaction prevents premature convergence and allows GWO to identify high-quality solutions within a reasonable timeframe.

3) **Fast Convergence**: Particularly valuable for computationally expensive simulations, GWO demonstrates a strong tendency to converge to optimal solutions quickly. This translates to significant time and resource savings, especially when dealing with complex problems involving numerous design variables. In the context of electromagnetics, for instance, GWO can significantly expedite the design process for antennas and other electromagnetic components.
4) **High Precision**: GWO not only finds optimal solutions rapidly but also exhibits a strong capability of achieving high precision. This translates to well-designed components with optimal performance characteristics. In antenna design optimization, for example, GWO can effectively optimize parameters to achieve desired radiation patterns, gain, and efficiency for the antenna.
5) **Adaptability**: A key advantage of GWO lies in its adaptability to diverse optimization problems. This stems from its flexible fitness function. While the core GWO algorithm remains the same, the fitness function can be readily customized to address specific challenges in various fields. This user-friendliness and minimal parameter tuning make GWO an attractive option for researchers and engineers working on a wide range of optimization tasks.
6) **Robustness**: GWO demonstrates a strong ability to avoid getting trapped in local minima (suboptimal solutions) within the search space. This robustness is attributed to the inherent exploration-exploitation balance of the algorithm. By continuously exploring new regions while refining promising solutions, GWO enhances the likelihood of finding the true global optimum.
7) **Potential for Parallelization**: GWO can be readily adapted for parallel computing environments. This allows for significant speedup in solving complex problems that require intensive computations. This potential for parallelization makes GWO particularly well-suited for large-scale optimization tasks in various domains, such as material science, signal processing, and machine learning.
8) **Future Directions**: Despite its impressive capabilities, GWO research continues to explore avenues for further improvement. Promising areas include:

   - **Hybrid Approaches**: Combining GWO with other optimization techniques like Particle Swarm Optimization (PSO) could leverage the strengths of both for even better performance.
   - **Problem-Specific GWO Variants**: Developing specialized GWO variants tailored to address specific classes of optimization problems could further enhance its effectiveness.
   - **Theoretical Analysis**: Further research into the theoretical underpinnings of GWO's performance can provide deeper insights and guide future algorithm development.

In conclusion, the Grey Wolf Optimizer has established itself as a powerful and versatile optimization tool. Its clear mathematical framework, efficient search process, and adaptability make it a valuable asset for researchers and engineers across a wide range of scientific and engineering disciplines. As research into GWO continues to evolve, its potential to tackle even more complex optimization challenges is certain to grow, solidifying its position as a cornerstone of the optimization toolbox.

References

[1] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965997813001853

[2] F. B. Ozsoydan, "Effects of dominant wolves in grey wolf optimization algorithm," *Applied Soft Computing*, vol. 83, p. 105658, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494619304387

[3] K. Deb, *Multi-Objective Optimization*. Boston, MA: Springer US, 2005, pp. 273–316. [Online]. Available: https://link.springer.com/book/10.1007/0-387-28356-0

[4] A. Khaghani, M. Meshkat, and M. Parhizgar, "A novel grey wolf optimizer for solving optimization problems," in *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 2019, pp. 1–6.

[5] M. Alizadeh, S. E. Mousavi, M. T. H. Beheshti, and A. Ostadi, "Combination of feature selection and hybrid classifier as to network intrusion detection system adopting fa, gwo, and bat optimizers," in *2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 2021, pp. 1–7.

[6] H. Q. Zheng, Y. T. Huang, and M. S. Tong, "A novel design of microwave absorbers based on multilayered composite materials for reduction of radar cross section," *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 37, no. 03, p. 326–334, Jul. 2022. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/article/view/11293

[7] H. Yang, Z. Guo, X. Li, Y. Zhang, X. Song, and S. Wang, "A compact circularly polarized crossed dipole antenna with wide bandwidth using split ring resonator and parasitic patches," *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 38, no. 04, p. 224–230, Aug. 2023. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/article/view/19043

[8] M. Sarabi and W. F. Perger, "A novel comb-line leaky-wave antenna for 77 ghz fmcw automotive radar system and comparison with buttler beamformers," *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 37, no. 09, p. 1005–1013, Feb. 2023. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/article/view/14001

[9] S. Koziel and A. Pietrenko-Dabrowska, "On nature-inspired design optimization of antenna structures using variable-resolution em models," *Scientific Reports*, vol. 13, no. 1, p. 8373, May 2023. [Online]. Available: https://doi.org/10.1038/s41598-023-35470-4

[10] S. Koziel, A. Pietrenko-Dabrowska, and B. Pankiewicz, "On accelerated metaheuristic-based electromagnetic-driven design optimization of antenna structures using response features," *Electronics*, vol. 13, no. 2, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/2/383

[11] T. Ding, H. Yang, Z. Chen, Z. Lin, B. He, and C.-Z. Han, "A 4-element uwb mimo antenna system with high isolation performance," *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 38, no. 10, p. 799–806, Mar. 2024. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/article/view/23049

[12] I. Bouchachi, A. Reddaf, M. Boudjerda, K. Alhassoon, B. Babes, F. N. Alsunaydih, E. Ali, M. Alsharef, and F. Alsaleem, “Design and performances improvement of an UWB antenna with DGS structure using a grey wolf optimization algorithm,” *Heliyon*, vol. 10, no. 5, p. e26337, Feb. 2024.
[13] L. Bai, Y. Zhuang, and Z. Zeng, “A highly compact sext-band bandpass filter with simple structure and wide bandwidth,” *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 37, no. 10, p. 1089–1095, Mar. 2023. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/article/view/12819
[14] L. Brezočnik, I. Fister, and V. Podgorelec, “Swarm intelligence algorithms for feature selection: A review,” *Applied Sciences*, vol. 8, no. 9, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/9/1521
[15] W. Awan, A. Zaidi, M. Hussain, N. Hussain, and I. Syed, “The design of a wideband antenna with notching characteristics for small devices using a genetic algorithm,” *Mathematics*, vol. 9, p. 2113, 09 2021.
[16] W. A. E. Ali, A. A. Ibrahim, and A. E. Ahmed, “A band stop to band pass filter transformation utilizing 3-d printing technique for c-band applications,” *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 37, no. 11, p. 1146–1152, Apr. 2023. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/article/view/18067
[17] J. Denies, “Metaheuristics for topology optimization : application to the design of electromagnetic devices,” 09 2013.
[18] A. A. Abbas and B. S. Samet, “A compact high gain wideband metamaterial antenna for sub-6 ghz applications,” *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 37, no. 08, p. 886–892, Jan. 2023. [Online]. Available: https://journals.riverpublishers.com/index.php/ACES/article/view/16033
[19] X. Li and K. M. Luk, “The grey wolf optimizer and its applications in electromagnetics,” *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 3, pp. 2186–2197, 2020.
[20] Y. Pan, J. Dong, M. Wang, and H. Luo, “Design of tunable ultra-wideband metasurface absorber with pixelated checkerboard pattern based on bgwo,” *Results in Physics*, vol. 57, p. 107376, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211379724000585
[21] J. Noh, Y.-H. Nam, S. So, C. Lee, S.-G. Lee, Y. Kim, T.-H. Kim, J.-H. Lee, and J. Rho, “Design of a transmissive metasurface antenna using deep neural networks,” *Opt. Mater. Express*, vol. 11, no. 7, pp. 2310–2317, Jul 2021. [Online]. Available: https://opg.optica.org/ome/abstract.cfm?URI=ome-11-7-2310