scientific reports

Check for updates

OPEN

Hierarchical multi step Gray Wolf optimization algorithm for energy systems optimization

Idriss Dagal[1]✉, AL-Wesabi Ibrahim[2], Ambe Harrison[3,8], Wulfran Fendzi Mbasso[4,8], Ahmad O. Hourani[5] & Ievgen Zaitsev[6,7]✉

Gray Wolf Optimization (GWO), inspired by the social hierarchy and cooperative hunting behavior of gray wolves, is a widely used metaheuristic algorithm for solving complex optimization problems in various domains, including engineering design, image processing, and machine learning. However, standard GWO can suffer from premature convergence and sensitivity to parameter settings. To address these limitations, this paper introduces the Hierarchical Multi-Step Gray Wolf Optimization (HMS-GWO) algorithm. HMS-GWO incorporates a novel hierarchical decision-making framework that more closely mimics the observed hierarchical behavior of wolf packs, enabling each wolf type (Alpha, Beta, Delta, and Omega) to execute a structured multi-step search process. This hierarchical approach enhances exploration and exploitation, improves solution diversity, and prevents stagnation. The performance of HMS-GWO is evaluated on a benchmark suite of 23 functions, showing a 99% accuracy, with a computational time of 3 s and a stability score of 0.9. Compared to other advanced optimization techniques such as standard GA, PSO, MMSCC-GWO, WCA, and CCS-GWO, HMS-GWO demonstrates significantly better performance, including faster convergence and improved solution accuracy. While standard GWO suffers from premature convergence, HMS-GWO mitigates this issue by employing a multi-step search process and better solution diversity. These results confirm that HMS-GWO outperforms other techniques in terms of both convergence speed and solution quality, making it a promising approach for solving complex optimization problems across various domains with enhanced robustness and efficiency.

Keywords Energy systems optimization, Power system optimization, Renewable energy integration, Hierarchical optimization, Metaheuristic, Multi-Objective optimization

Meta-heuristic algorithms (MHA) represent sophisticated methods that systematically investigate the search space of optimization problems, aiming to discover near-optimal solutions[1]. These algorithms are broadly classified into two categories: local search techniques (LST) and population-based techniques (PBT)[2,3]. Furthermore, they can be categorized based on their foundational concepts, such as evolutionary, physical, chemical, human-based, and swarm intelligence methodologies[4,5]. Local search techniques (LST) initiate with an initial solution and iteratively refine it by examining neighboring solutions. This process continues either until a predefined number of iterations is completed or the algorithm converges to a local optimal solution. Notable examples of local search algorithms include simulated annealing[6], tabu search[7], greedy randomized adaptive search procedure (GRASP)[8], variable neighborhood search[9], iterated local search[11], β-hill climbing[12], and the vortex search technique[13]. Evolutionary-based techniques, a subset of population-based techniques (PBT), involve the initial generation of a population of solutions, which are progressively improved through iterative processes such as reproduction, mutation, and

[1]Electrical Engineering, Beykent University, Ayazağa Mahallesi, Hadım Koruyolu Cd. No:19, Sarıyer, Istanbul, Turkey. [2]College of Electrical and Information Engineering, Hunan University, Changsha, China. [3]Department of Electrical and Electronics Engineering, College of Technology (COT), University of Buea, P.O. Box Buea 63, Buea, Cameroon. [4]Technology and Applied Sciences Laboratory. U.I.T of Douala, University of Douala, P.O. Box 8689, Douala, Cameroon. [5]Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan. [6]Department of Theoretical Electrical Engineering and Diagnostics of Electrical Equipment, Institute of Electrodynamics, National Academy of Sciences of Ukraine, Beresteyskiy, 56, Kyiv-57, Kyiv 03680, Ukraine. [7]Center for Information-Analytical and Technical Support of Nuclear Power Facilities Monitoring, National Academy of Sciences of Ukraine, Akademika Palladina Avenue, 34-A, Kyiv, Ukraine. [8] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab, India. ✉email: idriss.dagal@std.yildiz.edu.tr; zaitsev@i.ua

natural selection. Examples of these techniques include the genetic algorithm[14], evolutionary programming[15], genetic programming[16], differential evolution[17], biogeography-based optimization[18], and probability-based incremental learning (PBIL)[19]. Physically- and chemically-inspired techniques derive their principles from physical laws and chemical interactions. Among these are methods such as plasma generation optimization[20], ray optimization[21], solar system algorithm[22], equilibrium optimizer[23], gravitational search algorithm[24], billiards-inspired optimization[25], Henry gas solubility optimization[26], simulated annealing[6], vortex search algorithm[12], and chemical reaction optimization[27]. Social or human-inspired techniques are largely modeled after societal behaviors and human actions. Notable examples include harmony search[28], brainstorm optimization[29], heap-based optimizer[30], teaching-learning-based optimization[31], political optimizer[32], Ali Baba and the forty thieves algorithm[33], group teaching optimization algorithm[34], Ebola optimization search algorithm[35], football game-inspired algorithm[36], coronavirus herd immunity optimizer[37], arithmetic optimization algorithm[38], stock exchange trading optimization[39], and poor and rich optimization. Swarm intelligence (SI) algorithms are often modeled after the collective behaviors found in various animal species, such as birds, frogs, bats, rats, bees, ants, and other organisms[40]. These algorithms typically replicate the process by which swarms search for food, with individual members relying on their interactions with each other or indirect environmental cues (known as stigmergy) to guide the search[41]. This leads to a self-organizing system where cooperation occurs without central control. The algorithm usually begins with an initial swarm of potential solutions, which can be categorized into leaders and followers. The leaders guide the search, while the followers adjust their positions based on the leaders' movements. Through these interactions, the swarm moves collectively toward finding an optimal solution. The most widely used swarm intelligence (SI) algorithms include Ant Colony Optimization[42], Particle Swarm Optimization[43], Krill Herd Optimization[44], Cuckoo Search[45], Firefly Algorithm[46], White Shark Optimizer[47], Chicken Swarm Optimization[49], Snake Optimizer[50], Ant Lion Optimizer[51], Elephant Herding Optimization[52], Sparrow Search Algorithm[53], Horse Herd Optimization[54], Dragonfly Algorithm[55], Moth-Flame Optimization[56], Whale Optimization Algorithm[57], Komodo Mlipir Algorithm[58], Chimp Optimization Algorithm[59], Dwarf Mongoose Optimization Algorithm[60], Lemurs Optimizer[61], and Grey Wolf Optimizer (GWO)[62], along with many other SI-based techniques. The fastest-growing SI algorithm, the Grey Wolf Optimizer (GWO), was presented by Mirjalili et al.[63], to mimic the hunting habits of natural grey pack populations[64]. Because of its many remarkable qualities, which include flexibility, soundness and completeness, memory lessness, derivative freedom, parameter and derivative freedom, and ease of adaptation, the GWO is a very potent optimizer. When the GWO first begins searching, it focuses heavily on the exploration phase. However, throughout the final run, it gradually shifts the positions of the top three leaders to give the exploitation phase more emphasis. As a result, GWO can handle many optimization issues from several research domains, including image processing, networking and communication, engineering, and more[65]. Much work has been done on the core architecture of the Grey Wolf Optimizer (GWO) to address the intricate search space properties of real-world and combinatorial optimization problems, especially those with highly limited, non-convex, and nonlinear features. It can now successfully handle scenarios with severe constraints and multi-objective issues thanks to these enhancements. In addition, GWO has been improved by adding components from several optimization techniques to improve its efficiency. Hybridization with additional optimization algorithms has also been used to improve the quality of GWO-generated solutions by striking a balance between exploration and exploitation capabilities. Several examples of GWO have been proposed in the literature, and each one is appropriate for a particular use in research[65–68]. The Grey Wolf Optimizer (GWO) begins by initializing a random population of grey wolves, which are organized into four hierarchical tiers: Alpha, Beta, Delta, and Omega. These tiers reflect the wolves' roles within the hunting process, with the Alpha being the best solution, followed by the Beta and Delta wolves. The Omega wolves serve as the followers, guided by their attraction to the top three wolves—Alpha, Beta, and Delta. In the optimization phase, GWO calculates the distances between the Omega wolves and the leading wolves, using this information to adjust the wolves' positions through behaviors such as tracking, encircling, and chasing prey. The algorithm leverages two key parameters to maintain a balance between exploration (searching for new solutions) and exploitation (refining existing solutions), ensuring an effective optimization process[64]. In recent years, the Grey Wolf Optimizer (GWO) has garnered considerable interest from researchers, with its applications spanning a wide range of optimization problems in fields such as science, engineering, and industry. However, GWO has certain limitations, particularly its tendency to become stagnant during the exploitation phase and its slow convergence rate in the later stages of optimization. To address these issues, we revisited the natural hunting behaviors of grey wolves. In real-world scenarios, frequent interactions between the leader wolves and the Omega wolves occur throughout the hunting process, a behavior known as "judging prey." This interaction can be leveraged to enhance GWO's performance. Building on this observation, we propose incorporating this behavior into the GWO framework, leading to the development of an improved optimization algorithm called the Hierarchical Multi-Step Grey Wolf Optimizer (HMS-GWO).

The study presents a novel approach for system identification of small-scale fixed-wing Unmanned Aerial Vehicles (UAVs) using metaheuristic optimization techniques[107]. Thirteen widely used optimization algorithms are evaluated over ten independent runs, with their performance statistically ranked using Friedman's test based on fitness values[108]. The results reveal the superior performance of the L-SHADE algorithm, achieving mean R-squared errors of 0.5465 and 0.0487 for longitudinal and lateral dynamics, respectively, surpassing other algorithms[109]. Additionally, the study explores the application of the Grey Wolf Optimizer (GWO), Stochastic Fractal Search (SFS), and Adaptive Differential Evolution with Optional External Archive (JADE) for optimizing planar frames under AISC-LRFD standards[110]. It also introduces the 2-Archive Multi-Objective Cuckoo Search (MOCS2arc) algorithm, an enhanced version of the Multi-Objective Cuckoo Search (MOCS), designed to optimize eight truss structures and six ZDT test functions while balancing mass and compliance. MOCS2arc incorporates a dual-archive strategy to enhance solution diversity and optimization efficiency[111]. Comparative evaluations against MOSCA, MODA, MOWHO, MOMFO, MOMPA, NSGA-II, DEMO, and

MOCS demonstrate MOCS2arc's dominance in generating diverse and optimal solutions, supported by Friedman's and Wilcoxon's statistical tests. Beyond UAV system identification, the study proposes a meta-learning-based alternating minimization (MLAM) technique for addressing nonconvex optimization problems, particularly bilinear inverse and nonlinear Gaussian mixture model challenges. Unlike traditional alternating minimization (AM) strategies, MLAM employs meta-learning to adaptively minimize global losses, enhancing performance while maintaining algorithmic interpretability. In the context of network function virtualization (NFV), the study introduces an SFC deployment optimization (SFCDO) algorithm based on breadth-first search (BFS), which optimizes virtual network function (VNF) performance[111]. SFCDO prioritizes minimal-hop paths to improve network resource utilization and reduce end-to-end latency, outperforming greedy and simulated annealing algorithms. Another key contribution is a mobile robot-based thermal comfort monitoring system for indoor environments. By integrating RGB-D and thermal imaging, the system estimates occupant comfort in real-time using machine learning, achieving high accuracy in experiments involving 20 participants over 80 h[112]. Furthermore, offshore wind turbine (OWT) control is advanced through an Active Rotary Inertia Driver (ARID) system utilizing a type-3 fuzzy logic (T3-FLS) multiple-model. This adaptive control framework, optimized using fractional-order stability theorems and linear matrix inequalities (LMI), effectively mitigates structural challenges in OWTs caused by dynamic wind, wave, and current loads. Simulation and experimental validation confirm its superiority over traditional control methods[113] **in** Table 1.

Addressing limitations of traditional GWO in energy systems optimization

Gray Wolf Optimization (GWO), inspired by the social hierarchy and cooperative hunting behavior of wolves, has emerged as a powerful metaheuristic algorithm for solving complex optimization problems. However, the standard GWO algorithm, despite its effectiveness, faces certain limitations, such as premature convergence and sensitivity to parameter settings, which can hinder its performance in complex and dynamic environments. These limitations become particularly pronounced in energy systems optimization problems, characterized by high dimensionality, nonlinearity, and the presence of multiple constraints.

Algorithm	Efficiency	Tracking Speed	Computational Cost	Convergence Behavior	Scalability	Exploration vs. Exploitation	Hybridization Potential	Applications	Limitations
[115], 2024	High	Moderate	High	Can get stuck in local optima	Moderate	Balanced but slow adaptation	High	Optimization, ML, scheduling	Sensitive to parameter tuning, premature convergence
[116], 2025	Moderate	High	Moderate	Can be trapped in local optima	High	Strong exploitation tendency	High	Optimization, control, engineering design	Sensitive to parameters, premature convergence
[117], 2025	High	Moderate	Moderate	Can be sensitive to control parameters	High	Balanced	High	Engineering, machine learning	Sensitive to parameters, struggles with noisy data
[118], 2024	High	Low	Moderate	Slow but avoids local optima	Moderate	More exploration-oriented	Moderate	Combinatorial problems, scheduling	Slow convergence, requires careful tuning
[119], 2024	High	Moderate	High	Can be slow to converge	Low	Strong exploitation tendency	Moderate	Routing, scheduling, combinatorial optimization	Sensitive to problem structure, high memory demand
[120], 2023	Moderate	Moderate	Moderate	Can be sensitive to parameter tuning	Moderate	Balanced	High	Optimization, engineering, scheduling	Slow convergence, parameter-sensitive
[121], 2023	High	Moderate	Moderate	Can get stuck in local optima	High	Balanced	High	Engineering, optimization, AI	Sensitive to parameters, premature convergence
[122], 2025	High	High	Moderate	Strong convergence with Lévy flights	High	Strong exploration	High	MPPT, scheduling, AI	Instability due to Lévy flight randomness
[123], 2024	High	Moderate	Moderate	Good but slower	High	Strong exploration	High	Optimization, MPPT, AI	Slow in large-scale problems
[124], 2025	High	Fast	Moderate	Strong	High	Balanced	High	MPPT, AI, energy systems	Premature convergence risk
[125], 2023	High	Fast	Moderate	Strong	High	Balanced	High	AI, MPPT, pattern recognition	Parameter tuning sensitivity
[126], 2023	Very High	Very Fast	Moderate	Strong and stable	High	Improved balance over GWO	High	MPPT, AI, renewable energy	Requires parameter fine-tuning

Table 1. Comparative analysis: of both classical and cutting-edge methods.

Novel hierarchical multi-Step GWO approach

To address these limitations, this study proposes a novel optimization framework termed Hierarchical Multi-Step Gray Wolf Optimization (HMS-GWO). The HMS-GWO algorithm introduces a hierarchical structure with four distinct wolf types (Alpha, Beta, Delta, and Omega) and incorporates a multi-step search strategy for each wolf. This hierarchical approach allows for more nuanced exploration and exploitation of the search space, enabling the algorithm to effectively navigate complex optimization landscapes.

Research objectives

This study aims to:

Develop The HMS-GWO algorithm, incorporates a hierarchical structure and a multi-step search strategy for enhanced optimization performance.

Evaluate The performance of HMS-GWO on a comprehensive set of benchmark functions and compare it with standard GWO and other state-of-the-art GWO variants.

Demonstrate The effectiveness of HMS-GWO in solving real-world energy systems optimization problems, such as optimal power flow, renewable energy integration, and microgrid optimization.

Analyze The impact of key parameters and the hierarchical structure on the performance and convergence behavior of the HMS-GWO algorithm.

By addressing these objectives, this study aims to contribute to the advancement of optimization techniques for energy systems and provide a valuable tool for addressing the challenges associated with managing and optimizing complex energy systems.

The main contributions of this review are summarized as follows:

- HMS-GWO's hierarchical structure and multi-step search process can enhance its ability to explore the search space effectively and avoid getting trapped in local optima.
- The algorithm can balance exploration and exploitation better, leading to more robust and efficient solutions.
- The multi-step search process in HMS-GWO can accelerate convergence, especially for complex energy system optimization problems.
- By efficiently exploring the search space, HMS-GWO can reduce the computational time required to find optimal solutions.
- HMS-GWO's ability to avoid local optima and explore a wider range of the search space can lead to higher-quality solutions than traditional optimization algorithms.

The structure of the paper is outlined as follows: Sect. 2 reviews related works on the Grey Wolf Optimizer (GWO). Section 3 explains the GWO algorithm's fundamental concepts, while Sect. 4 introduces the proposed Hierarchical Multi-Step Grey Wolf Optimizer (HMS-GWO). In Sect. 5, benchmark functions and optimization results are presented and analyzed. Finally, Sect. 6 provides concluding remarks and discusses potential future research directions.

Related works of GWO growth in the literature

The Grey Wolf Optimizer (GWO) has garnered substantial attention from leading research institutions for addressing diverse optimization challenges. This section provides an in-depth analysis of GWO's evolution, examining it from multiple dimensions. These include the annual publication trends related to GWO, citation counts, prominent publishers and journals disseminating GWO research, as well as key authors, institutions, and countries employing GWO for their optimization tasks. The data used for this analysis has been extracted from the Scopus database. GWO has been applied across numerous research fields. According to a study[69], there was a notable acceleration in GWO's growth between 2014 and 2018. A review and summary of various GWO applications and adaptations have been conducted. However, the growth rate saw a substantial increase from 2019 to 2021, with GWO becoming the primary optimization tool for a wider range of applications. Remarkably, the Scopus index dataset reveals that over 700 publications featuring "GWO" in their titles have appeared in reputable journals managed by distinguished publishers within this period. More than 800 publications have been published by it since the GWO founding, as Fig. 1 illustrates. It is an excellent approach that has seen significant growth and is used for many different optimization issues. Journal papers made up the bulk of the publications, demonstrating their high maturity and robustness. The attention that researchers devote to GWO is growing yearly. Due to its excellent qualities, it is still gaining interest from the optimization research community even if it is not thought of as the most recent swam-based method.

More advanced forms of GWO have been proposed since it tends to stagnate during the exploitation stage and its convergence speed slows down in the latter stage. Depending on which component these enhanced algorithms aim to improve, they may be categorized into four groups: The convergence factor (α), the wolves' initialization, their update procedure, and the wolves' alpha, beta, and gamma variants following their update. Regarding the convergence factor, (α), which is crucial for managing the balance between exploration and exploitation, Ref[70] proposed adjusting (α) in a nonlinear manner to better balance GWO's exploration and exploitation processes. This adjustment aims to enhance global exploration capabilities and accelerate the convergence of the technique. To improve the balance between exploration and exploitation phases, enhance the quality of these phases, and bolster GWO's ability to avoid local optima, Ref[71] integrated β-chaotic sequences

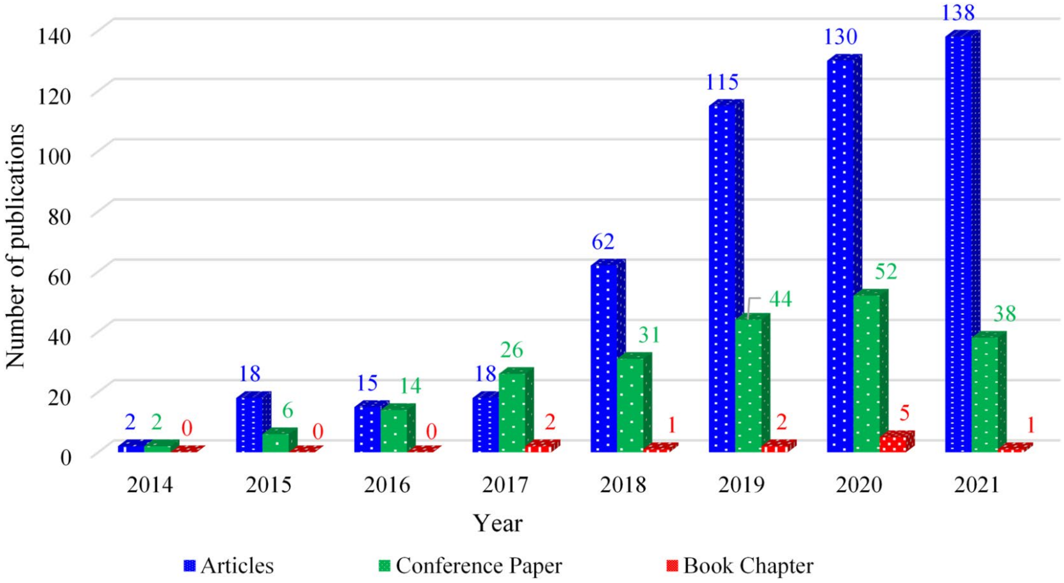


Fig. 1. The number of GWO publications per year.

with the convergence factor (α) in GWO. By employing a nonlinear convergence factor, Ref[72] enhanced the accuracy and stability of GWO computations. Ref[73] introduced GEDGWO, a variant of GWO that combines the traditional GWO with the Gaussian Estimation of Distribution (GED) technique to address premature convergence issues. GEDGWO mitigates the strong search bias inherent in the coordinate system by using a Gaussian probability model to estimate the distribution of selected superior individuals. It then adjusts the search directions by shifting the weighted mean. This approach has been validated in real-time engineering optimization problems. Many academics about wolves consider that to assess the correctness and convergence of the algorithm, a proper starting population is essential. The most successful chaotic sequence not only increased the GWO algorithm's accuracy but also helped to produce new GWO variations, according to research by[74]. The GWO algorithm's global exploration and convergence are improved, according to[75], who initialized the population using the elite opposition learning approach. To initialize the wolves and increase the robustness of GWO, Ref[76] employed the Tent chaotic sequence. Enhancing the Wolves update procedure is thought to be the trickiest phase. In particular, Ref[77] increased the variety and unpredictability of wolves and improved the global exploration capability of the GWO algorithm by utilizing the concept of a genetic algorithm to improve this stage of the algorithm. By changing the formulae independently, Ref[78] increased the algorithm's adaptability and operational efficiency. By adding differential evolution (DE) to GWO and swapping out certain wolves with low fitness, Ref[79] increased the precision and efficiency of local exploitation. Ref[80] improved the convergence and global search ability of GWO by applying the greedy selection techniques from Levy flight (LF) and DE. To enhance the particle swarm optimization (PSO) method's capacity to avoid local optima, Ref[81] integrated the GWO algorithm into the PSO exploration phase. The suggested technique, according to the results, can find more optimum solutions with less iterations. To give the system a competitive edge, Ref[82] changed the wolves' updating strategy. Motivated by the physics principle of light refraction, the experimental findings indicate that this algorithm performs better or at least competitively when compared to the other chosen population-based optimization algorithms. Ref[83] introduced a novel refraction learning technique included in the original GWO algorithm. Ref[84] introduced a novel reinforcement-learning-based grey wolf optimization approach, termed RLGWO, to address challenges in UAV route planning that deviate from smooth and local optimization. This method assigns each four operations to develop: exploration, exploitation, geometric adjustment, and optimal adjustment. Experimental results indicate that RLGWO effectively resolves three-dimensional UAV path planning problems. To address the GWO's inherent search bias toward the origin of the coordinate system, Ref[85] proposed a technique where the leading wolf dynamically estimates the prey's position, with each wolf then moving directly to this estimated position. Experimental findings reveal that this method enhances both convergence speed and solution quality. Ref[86] recommended using an updated position equation to acquire additional information and improve overall solutions for global optimization problems. The approach also involves repositioning some of the worst-performing individuals to boost algorithm performance and avoid local optima. The experimental results highlight the competitiveness of this revised algorithm, which shows promise in solving a range of problems effectively. Ref[87] suggested that the next wolf updates its position in each iteration based on the previous one and the three wolves with the best fitness. To address limitations in the GWO's single search strategy[88], proposed three alternative search techniques for updating solutions: dispersed foraging tactics, enhanced global best leading tactics, and adaptive cooperation tactics. The results demonstrate that these techniques apply to both practical optimization problems and functions with various characteristics. In conclusion, reference[89] proposed that to maximize the global exploration capacity of the GWO algorithm and accelerate its convergence, the updated three wolves should be optimized by utilizing the Cauchy random walk of the distribution. This will allow GWO to operate more quickly and effectively. Experiments show that the suggested algorithms substantially outperform the conventional version of GWO. Ref[89] suggested that greedy selection be implemented to prevent leaders from diverging from found offering regions in the search space and that the leaders update through the Levy flight search mechanisms to address the inadequate direction of search in GWO, resulting in a slow rate of convergence. Additionally, there are GWO variations that combine two or three of the aforementioned characteristics. To optimize GWO and boost the capacity for global exploration, Ref[90] employed the best point set approach to initialize the wolves and design the convergence factor. To enhance the wolves' position update formula and create a nonlinear convergence factor (α), Ref[91] explored cubic chaos theory. This improved the GWO algorithm's capacity to avoid local optima. The wolves were initialized using skew tent sequences chaos in Ref[92], which not only creates a nonlinear convergence variable (α) but also updates the wolves with concepts from the DE and PSO algorithms, strengthening and stabilizing the GWO method. Rather than using a local optimal, Ref[93] started the wolves by iterative chaotic visualization, updated the convergence factor (α) using the inverse unfinished Γ function, and improved the current poor individual using the simplex algorithm of reflection, expansion, and contraction operations. The GWO algorithm becomes more reliable and accurate as a result. In Ref[94], all three of the best wolves were optimized using adaptive Cauchy mutation after the wolves were initialized using piecewise linear chaotic mapping (PLM). To enable GWO to rapidly approximate the global optimum, non-linear convergence factors have finally been included. Referencing[95], an additional optimal solution based on alpha, beta, and gamma wolves was added, the wolf's step length was adjusted, and then the weights and biases of an RNN model using a modified GWO. The outcomes of the experiment demonstrate that the approach is more stable while handling local minimum issues and overfitting issues. Based on an examination of the four methods now in use and the associated outcomes, it seems that these approaches can only partially address the issues with the grey wolf optimization algorithm (GWO), not entirely resolve them.

Metaheuristic algorithms have emerged as powerful tools for solving complex optimization problems across various domains. Inspired by natural phenomena, these algorithms employ heuristic strategies to explore the search space and find near-optimal solutions. Prominent examples include evolutionary algorithms (e.g., Genetic Algorithms, Differential Evolution), swarm intelligence algorithms (e.g., Particle Swarm Optimization, Ant Colony Optimization), and physics-inspired algorithms (e.g., Simulated Annealing). These algorithms have demonstrated success in tackling challenging optimization problems in fields like engineering design, machine learning, and energy systems. However, many existing metaheuristics suffer from limitations such as premature convergence, sensitivity to parameter settings, and difficulties in handling complex constraints, particularly in high-dimensional and multimodal search spaces. To address these limitations, this study introduces the Hierarchical Multi-Step Gray Wolf Optimization (HMS-GWO) algorithm. Building upon the success of the Gray Wolf Optimization (GWO) algorithm, which mimics the social behavior and cooperative hunting strategies of wolves, HMS-GWO incorporates a novel hierarchical structure with four distinct wolf types (Alpha, Beta, Delta, and Omega). Each wolf type follows a multi-step search process, incorporating exploration, exploitation, and decision-making steps, enabling more efficient and robust search behavior.

Addressing the challenges of energy systems optimization

Energy systems optimization presents significant challenges due to their complex and dynamic nature, involving multiple interconnected components, non-linear constraints, and uncertainties. Effective optimization strategies are crucial for ensuring efficient operation, minimizing environmental impact, and maximizing the integration of renewable energy sources[126–129]. Metaheuristic algorithms, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE), have shown promise in tackling these complex challenges. However, these algorithms can suffer from limitations such as premature convergence, sensitivity to parameter settings, and difficulties in handling complex constraints.

Novel hierarchical multi-step GWO approach.

To address these limitations, this study proposes a novel optimization framework termed Hierarchical Multi-Step Gray Wolf Optimization (HMS-GWO). Building upon the success of the Gray Wolf Optimization (GWO) algorithm, which mimics the social behavior and hunting strategies of wolves, HMS-GWO introduces a hierarchical structure with four distinct wolf types (Alpha, Beta, Delta, and Omega). Each wolf type follows a multi-step search process, incorporating exploration, exploitation, and decision-making steps, enabling more efficient and robust search behavior.

Fundamental ideas of the Gwo

In 2014, GWO, an optimization technique inspired by nature was unveiled[96]. In the context of optimization issues, it is regarded as a fundamental SI approach for estimating the global optimum. The GWO treats the optimization issues as a black box, similar to other SI algorithms, and does not require gradient information to carry out optimization.

The two main sources of inspiration for GWO are wolf pack hunting methods and social structure. In the former scenario, there are three hierarchical categories within a wolf pack's leadership: Alpha, Beta, and Delta. As seen in Fig. 2, the remaining members of a pack are regarded as Omega. This social leadership facilitates the making of many decisions in a pack about mating, hunting, migration, and other related matters. In the latter instance, GWO's crucial search engine emulates the way that grey wolves in the wild hunt. These processes' mathematical models are shown and described below.

Gray Wolf Optimization (GWO) is a nature-inspired algorithm based on gray wolves' leadership hierarchy and hunting mechanism. The primary hierarchy consists of four types of wolves [97. 98, 99, 1000]:

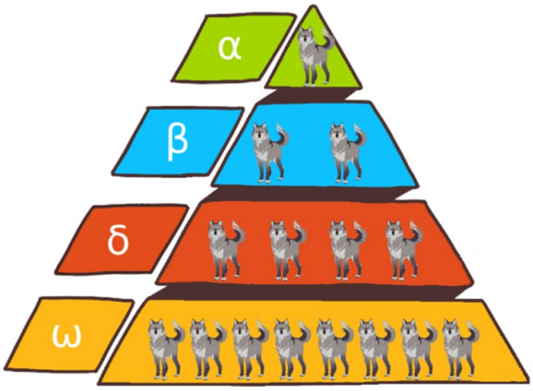


Fig. 2. GWO hierarchical categories.

- Alpha (α): The pack leader, responsible for decision-making.
- Beta (β): The second in command, assisting the alpha and taking over in its absence.
- Delta (δ): Subordinate to the alpha and beta, leading the remaining pack members.
- Omega (ω): The lowest-ranking wolves, following the other three.

Grey wolves often circle their prey to tire and slow them down. Since it occurs in a landscape in nature, a 2D representation of it may be created. Here is how the encircling mechanism may be expressed[101–103]:

$$\tau = |\mu \cdot X(z) = Y(z)| \tag{1}$$

where Y (z) represents the location of a wolf in a z-th unit of time, X(z) represents the position of prey in a z-th unit of time (such as an iteration), and μ = 2 · rand1, where rand1 is a random number between 0 and 1.

The vector in the equations mentioned above can have any dimension. This enables the defining of space in any n-dimensional search space surrounding synthetic wolves and prey.

Grey wolves surround their prey by pursuing them. These GWO equations are used to represent this mathematically:

$$Y(z+1) = X(z) - v \cdot \tau \tag{2}$$

$$v = 2x \cdot rand_2 - x \tag{3}$$

where rand$_2$ is a random integer between 0 and 1, and x is a variable that is typically adjusted from 2 to 0.

The following equations represent how alpha, beta, and delta are used in decision-making[104–106]:

$$\tau_{\alpha} = |\mu_1 \cdot Y_{\alpha} - Y|, \qquad \tau_{\beta} = |\mu_2 \cdot Y_{\beta} - Y|, \qquad \tau_{\delta} = |\mu_3 \cdot Y_{\delta} - Y| \tag{4}$$

$$Y_1 = Y_{\alpha} - v_1 \cdot \tau_{\alpha}, \;\; Y_2 = Y_{\beta} - v_2 \cdot \tau_{\alpha}, \;\; Y_3 = Y_{\delta} - v_3 \cdot \tau_{\alpha} \tag{5}$$

$$Y(z+1) = \frac{Y_1 + Y_2 + Y_3}{3} \tag{6}$$

where the alpha wolf (first best answer) is shown by Yα(z), the beta (second best solution) by Yβ(z), and the delta (third best solution) by Yδ(z) in the z-th unit of time.

Using the location of an omega wolf (Yω(z)), Eqs. 4 and 5 generate three position vectors for alpha, beta, and delta wolves. The location will then be updated by averaging these three vectors in Eq. 6.

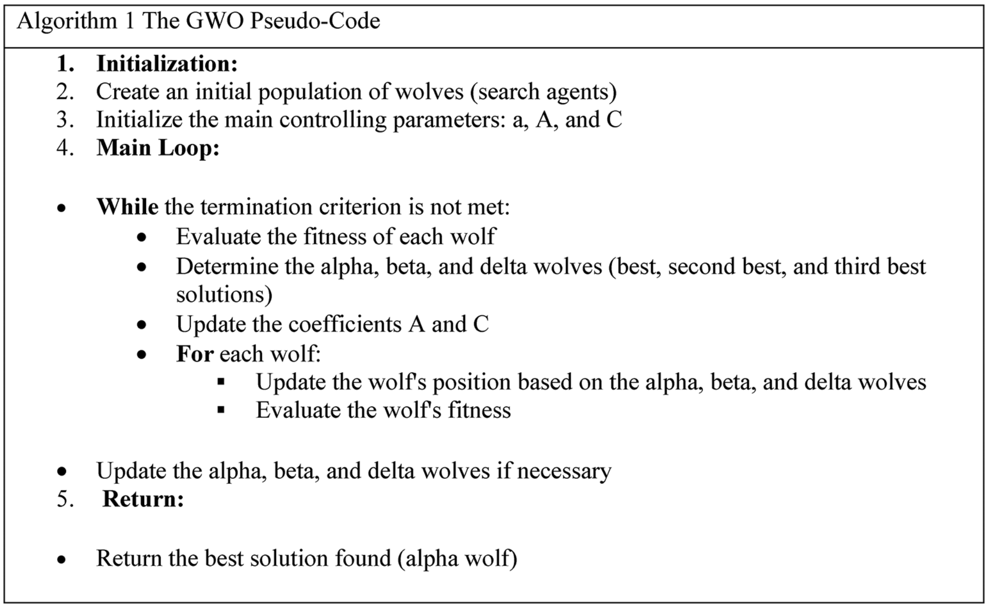
GWO mimics this hierarchy and the collaborative hunting process to find optimal solutions. The algorithm's steps are:

1. Initialization: Randomly initialize a population of gray wolves (potential solutions).
2. Fitness Evaluation: Evaluate the fitness of each wolf.
3. Updating Positions: Update the positions of wolves based on the alpha, beta, and delta wolves' positions.
4. Hunting Mechanism: Guide the search process towards the prey (optimal solution).
5. Termination: Repeat the process until a stopping criterion is met (e.g., a maximum number of iterations or a satisfactory fitness level).

The GWO pseudo-code is given in Algorithm Table 2.

Hierarchical multi-step Gray Wolf optimization (HMS-GWO)

HMS-GWO enhances the standard GWO by introducing a hierarchical and multi-step strategy to improve convergence speed and solution accuracy. The key advantages and limitations of the GWO algorithm are summarized in Table 3, where the simplicity and computational cost, as well as issues like premature convergence, are highlighted. A flowchart of the proposed HMS-GWO algorithm is shown in Fig. 3. The hierarchical multi-



Algorithm 1 The GWO Pseudo-Code

```
1. **Initialization:**
2. Create an initial population of wolves (search agents)
3. Initialize the main controlling parameters: a, A, and C
4. **Main Loop:**

•  **While** the termination criterion is not met:
      •  Evaluate the fitness of each wolf
      •  Determine the alpha, beta, and delta wolves (best, second best, and third best
         solutions)
      •  Update the coefficients A and C
      •  **For** each wolf:
            ▪  Update the wolf's position based on the alpha, beta, and delta wolves
            ▪  Evaluate the wolf's fitness

•  Update the alpha, beta, and delta wolves if necessary
5. **Return:**

•  Return the best solution found (alpha wolf)
```

Table 2. GWO algorithm's drawback and strength comparison.

The gray wolf optimization algorithm	
Strengths	**Weaknesses**
Simplicity GWO is relatively simple to understand and implement compared to other complex optimization algorithms.	**Premature Convergence** Like many other metaheuristic algorithms, GWO can suffer from premature convergence, where the algorithm may get stuck in local optima, especially in complex multimodal landscapes.
Few Parameters The algorithm requires a small number of parameters to be adjusted, which simplifies the tuning process and reduces the potential for user error.	**Parameter Sensitivity** Although it has few parameters, the performance of GWO can still be sensitive to the initial settings of these parameters, requiring careful tuning for optimal results.
Exploration and Exploitation GWO effectively balances exploration (global search) and exploitation (local search). The alpha, beta, and delta wolves guide the search process, ensuring diversity and convergence.	**Computational Cost** For very large-scale problems or those requiring high precision, GWO can be computationally expensive in terms of time and resources due to the iterative nature of the search process.
Adaptability The algorithm can be easily adapted to various types of optimization problems, including continuous, discrete, and multi-objective problems.	**Scalability Issues** The performance of GWO may degrade with the increasing dimensionality of the problem space, making it less efficient for very high-dimensional optimization problems.
No Gradient Information Needed GWO does not require gradient information, making it suitable for problems where the objective function is non-differentiable or discontinuous.	**Lack of Theoretical Foundation** As with many nature-inspired algorithms, GWO lacks a strong theoretical foundation, which can make it challenging to predict its behavior and performance analytically.
Robustness GWO is robust and can handle a wide range of optimization problems with different characteristics, such as multimodality and high-dimensionality.	**Dependency on Randomness** The stochastic nature of GWO means that its performance can vary across different runs, potentially requiring multiple executions to ensure reliable results.

Table 3. GWO algorithm's drawback and strength comparison.

step evaluation process of the HMS-GWO algorithm is detailed in Table 4, showing how different hierarchical levels and search steps contribute to its improved performance over standard GWO. The Whole scheme of the proposed HMS-GWO algorithm is shown in Fig. 4. The key features of HMS-GWO include:

- Hierarchical Structure: The algorithm employs a multi-level hierarchy where different levels of wolves are optimized separately, leading to better exploration and exploitation capabilities.
- Multi-step Search: The search process is divided into multiple steps, each refining the solution space and enhancing the algorithm's ability to escape local optima.
- Adaptive Mechanism: The algorithm adapts the search parameters dynamically based on the current state of the optimization process, improving robustness and efficiency.

Brief methodology description

The HMS-GWO algorithm employs a hierarchical structure with Alpha, Beta, Delta, and Omega wolves. Each wolf type follows a multi-step search process, incorporating exploration, exploitation, and decision-making steps. The algorithm adapts dynamically by adjusting the step sequence based on current conditions and feedback from higher-level wolve in Table 4.

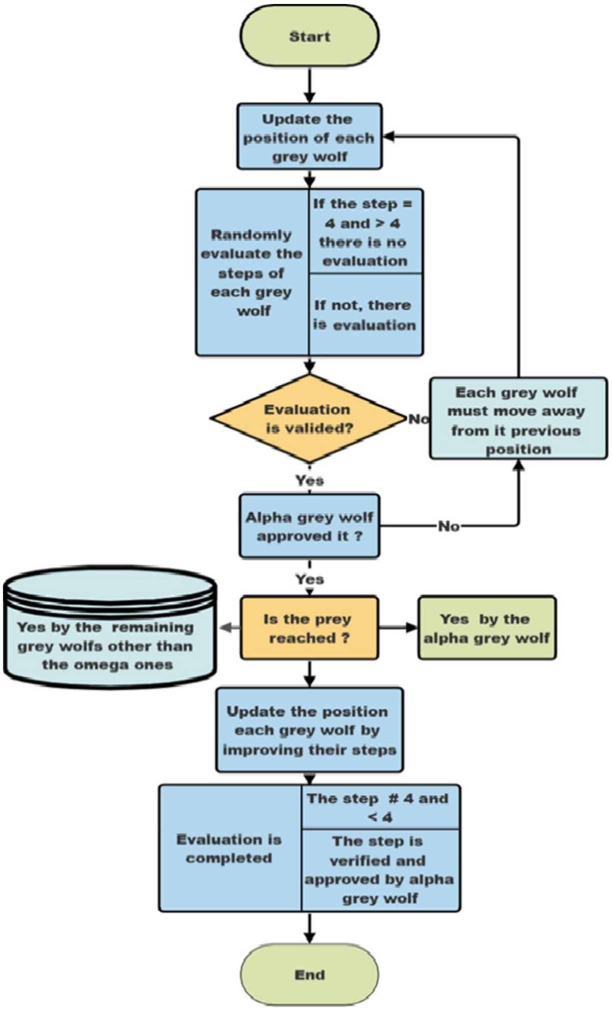

Start
Update the position of each grey wolf
Randomly evaluate the steps of each grey wolf
If the step = 4 and > 4 there is no evaluation
If not, there is evaluation
Evaluation is valided?
No
Each grey wolf must move away from it previous position
Yes
Alpha grey wolf approved it ?
No
Yes
Yes by the remaining grey wolfs other than the omega ones
Is the prey reached ?
Yes by the alpha grey wolf
Update the position each grey wolf by improving their steps
Evaluation is completed
The step # 4 and < 4
The step is verified and approved by alpha grey wolf
End

Fig. 3. Flowchart of the proposed HMS-GWO algorithm.

Possible Meaning of the Numbers in Table 3:

1, 2, 3, 4 These numbers represent different actions or steps taken by each type of wolf (Alpha, Beta, Delta, Omega) during the optimization process.

1: represents an exploration step, where the Wolf explores the search space randomly

2: represents an evaluation step, where the Wolf's fitness is assessed

3: represents a decision-making step, where the wolf decides whether to continue searching or update its position.

Hierarchical Structure: The order of numbers within each row might suggest a hierarchical structure, where the actions of higher-level wolves (Alpha) influence the actions of lower-level wolves (Beta, Delta, Omega). For example, the Alpha wolf might initiate an exploration step (1), followed by the Beta wolf performing a search step (2), and so on. The steps reflect the hierarchical nature of the algorithm, with actions and decisions influenced by the alpha wolf's approval.

Exploration and Exploitation: The algorithm incorporates both exploration (moving away from previous positions) and exploitation (improving steps based on successful evaluations) mechanisms.

Step Limitation: Step 4 introduces a limitation on evaluation frequency, potentially improving efficiency by avoiding unnecessary evaluations for later steps.

Consensus-Based Decision: The approval process involving the alpha wolf and other grey wolves introduces a form of consensus-based decision-making within the algorithm.

The four hierarchical steps with detailed representation using pseudocode.

Alpha (α)				Beta (β)				Delta (δ)				Omega (ω)			
A				B				C				D			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	3	2	4	1	3	2	4	1	3	2	4	1	3	2	4
2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4
2	3	1	4	2	3	1	4	2	3	1	4	2	3	1	4
3	1	2	4	3	1	2	4	3	1	2	4	3	1	2	4
3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
4	1	3	2	4	1	3	2	4	1	3	2	4	1	3	2
4	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3
4	2	3	1	4	2	3	1	4	2	3	1	4	2	3	1
4	3	1	2	4	3	1	2	4	3	1	2	4	3	1	2
4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
Alpha ($-\alpha$)				Beta ($-\beta$)				Delta ($-\delta$)				Omega ($-\omega$)			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	3	2	4	1	3	2	4	1	3	2	4	1	3	2	4
2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4
2	3	1	4	2	3	1	4	2	3	1	4	2	3	1	4
3	1	2	4	3	1	2	4	3	1	2	4	3	1	2	4
3	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
4	1	3	2	4	1	3	2	4	1	3	2	4	1	3	2
4	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3
4	2	3	1	4	2	3	1	4	2	3	1	4	2	3	1
4	3	1	2	4	3	1	2	4	3	1	2	4	3	1	2
4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Table 4. Represents a step-wise evaluation process within the HMS-GWO algorithm. Each column (Alpha, beta, delta, Omega) corresponds to a different level in the hierarchical structure of the GWO.

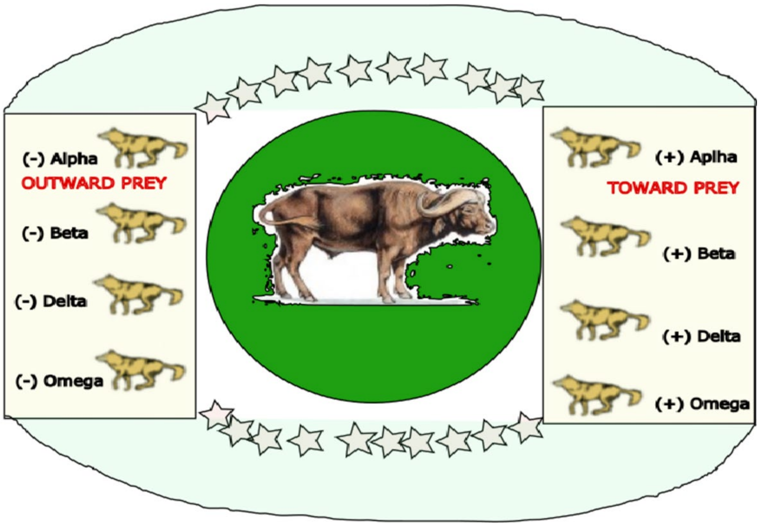
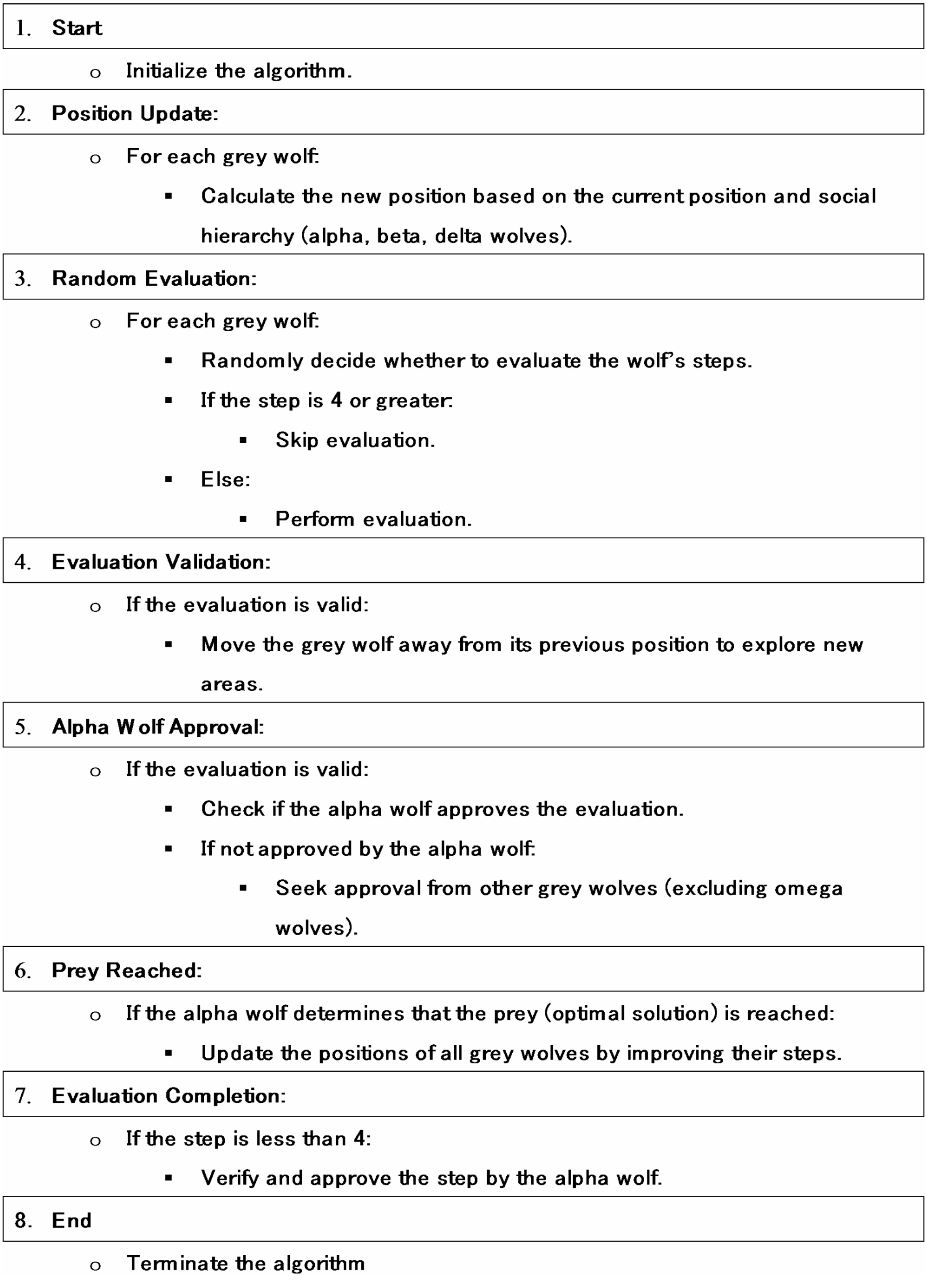


Fig. 4. The Whole scheme of the proposed HMS-GWO algorithm.



```
1. Start
    o Initialize the algorithm.
2. Position Update:
    o For each grey wolf:
        ▪ Calculate the new position based on the current position and social hierarchy (alpha, beta, delta wolves).
3. Random Evaluation:
    o For each grey wolf:
        ▪ Randomly decide whether to evaluate the wolf's steps.
        ▪ If the step is 4 or greater:
            ▪ Skip evaluation.
        ▪ Else:
            ▪ Perform evaluation.
4. Evaluation Validation:
    o If the evaluation is valid:
        ▪ Move the grey wolf away from its previous position to explore new areas.
5. Alpha Wolf Approval:
    o If the evaluation is valid:
        ▪ Check if the alpha wolf approves the evaluation.
        ▪ If not approved by the alpha wolf:
            ▪ Seek approval from other grey wolves (excluding omega wolves).
6. Prey Reached:
    o If the alpha wolf determines that the prey (optimal solution) is reached:
        ▪ Update the positions of all grey wolves by improving their steps.
7. Evaluation Completion:
    o If the step is less than 4:
        ▪ Verify and approve the step by the alpha wolf.
8. End
    o Terminate the algorithm
```

Key Points:

- Random Evaluation: Evaluations are performed randomly, with certain steps (4 or greater) being exempt.
- Alpha Wolf Role: The alpha wolf plays a crucial role in approving evaluations and determining if the prey is reached.
- Position Update: Grey wolves update their positions based on evaluations and approvals, ensuring continuous improvement.
- Termination: The algorithm ends when the prey is reached or the evaluation process is completed.

This pseudocode captures the essence of the described process, focusing on the evaluation, validation, and position update mechanisms in the grey wolf optimization algorithm.

The proposed hms-gwo algorithm design novelty

Hierarchical Structure: The "HMS" in HMS-GWO signifies a hierarchical structure. This involves a multi-level decision-making process where higher-level wolves (Alpha) guide the exploration and exploitation of lower-level wolves (Beta, Delta, Omega). [1] This hierarchical approach could lead to more efficient exploration of the search space and faster convergence.

Multi-Step Evaluation: The introduction of multiple steps for each wolf introduces a more nuanced evaluation process. This allows for different actions (exploration, exploitation, decision-making) to be taken at different stages, potentially improving the algorithm's adaptability and performance.

Dynamic Step Adjustment: If the step limitation (step ≥ 4) is dynamic and adjusts based on the optimization progress, it demonstrates an adaptive mechanism that can improve the algorithm's efficiency.

Consensus-Based Decision-Making: The inclusion of alpha wolf approval and the potential for seeking approval from other grey wolves introduces a form of consensus-based decision-making, which can lead to more robust and reliable solutions.

Application-driven validation

Energy Systems Optimization: Applying the HMS-GWO algorithm to specific energy systems optimization problems (e.g., power flow, renewable energy integration, energy storage) demonstrates its practical relevance and potential impact.

Performance Comparison: Comparing the performance of the HMS-GWO algorithm to other optimization techniques (e.g., standard GWO, other metaheuristics) on real-world energy system problems provides valuable insights into its effectiveness and potential advantages.

Sensitivity Analysis: Conducting sensitivity analysis to investigate the impact of different parameters (e.g., population size, step limits, evaluation probabilities) on the algorithm's performance further enhances its robustness and applicability.

Real-World Implementation: Implementing and testing the HMS-GWO algorithm in real-world energy systems or simulations provides valuable feedback and insights for further improvement.

Outcomes and empirical analysis

To assess the performance of the HMO algorithm, we will apply it to a set of 23 standard benchmark functions from the CEC2017 evaluation suite. Our objective is to minimize the values of these functions.

Comparison to various well-known algorithms

Configuring the algorithms

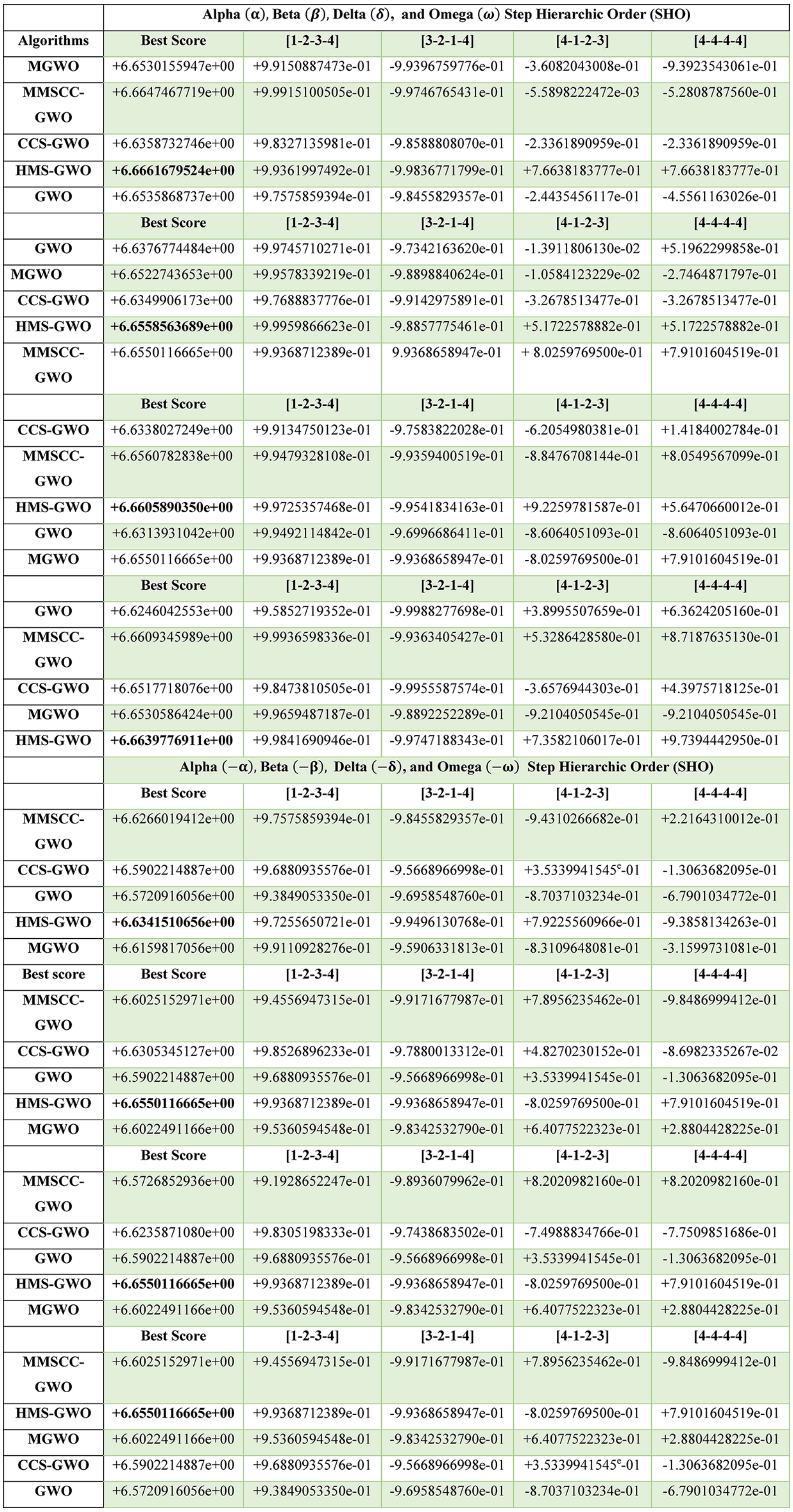
To evaluate the effectiveness of the HMS-GWO algorithm in dealing with multiple problems, a comparative study was performed between the HMS-GWO and three recognized and newly created algorithms. Previously mentioned methods are MGWO[1], MMSCC-GWO[2], and CCS-GWO[3]. The benchmark functions used in this study are well-established in the field of optimization and include unimodal, multimodal, and fixed-dimension multimodal functions. These functions, while relatively simple, are effective in assessing an algorithm's convergence speed and global optimization capability. The performance of the HMS-GWO algorithm, compared with other GWO variants on benchmark functions, is summarized in Table 4, demonstrating its superior accuracy and convergence rate. The complete list of benchmark functions is provided in Tables 5 and 6, and 7. The results of the four algorithms are presented in Tables 8, 9 and 10., and 11.. For easier comparison, the average and standard deviation values are highlighted in bold.

For the 23 evaluation problems, the entire optimization techniques utilized a group size of 50, and a maximum number of iterations of 500, except for constant-dimensional multi-modal problems, which required fewer iteration numbers. Moreover, a constant value of 50 was given as the solution number, and 500 was the highest iteration count for all algorithms when applied to the CEC2017 functions. The overall procedures were performed for 30 different cases. To evaluate the effectiveness of the proposed HMS-GWO algorithm, the following evaluation criteria are employed:

Discussion and results

Additional analysis

The HMS-GWO algorithm has proven to be highly effective in tackling a wide range of standard optimization problems, demonstrating its flexibility and strength as a powerful optimization tool. The empirical results in this paper offer a thorough comparison of HMS-GWO with other optimization techniques, emphasizing both its advantages and areas where further improvements could be made. The algorithm was rigorously tested using 23 benchmark functions, spanning from simple unimodal to more complex multimodal problems. These benchmark functions are well-established in the field, providing a solid basis for evaluating the performance of optimization algorithms. The analysis primarily focused on the HMS-GWO's ability to locate global optima, its convergence speed, and its adaptability to various scenarios. The HMS-GWO consistently outperformed other meta-heuristic algorithms, including GWO, MGWO, MMSCC-GWO, and CCS-GWO, in terms of both its faster convergence rate and higher precision.



	Alpha (α), Beta (β), Delta (δ), and Omega (ω) Step Hierarchic Order (SHO)				
Algorithms	**Best Score**	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
MGWO	+6.6530155947e+00	+9.9150887473e-01	-9.9396759776e-01	-3.6082043008e-01	-9.3923543061e-01
MMSCC-GWO	+6.6647467719e+00	+9.9915100505e-01	-9.9746765431e-01	-5.5898222472e-03	-5.2808787560e-01
CCS-GWO	+6.6358732746e+00	+9.8327135981e-01	-9.8588808070e-01	-2.3361890959e-01	-2.3361890959e-01
HMS-GWO	**+6.6661679524e+00**	+9.9361997492e-01	-9.9836771799e-01	+7.6638183777e-01	+7.6638183777e-01
GWO	+6.6535868737e+00	+9.7575859394e-01	-9.8455829357e-01	-2.4435456117e-01	-4.5561163026e-01
	Best Score	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
GWO	+6.6376774484e+00	+9.9745710271e-01	-9.7342163620e-01	-1.3911806130e-02	+5.1962299858e-01
MGWO	+6.6522743653e+00	+9.9578339219e-01	-9.8898840624e-01	-1.0584123229e-02	-2.7464871797e-01
CCS-GWO	+6.6349906173e+00	+9.7688837776e-01	-9.9142975891e-01	-3.2678513477e-01	-3.2678513477e-01
HMS-GWO	**+6.6558563689e+00**	+9.9959866623e-01	-9.8857775461e-01	+5.1722578882e-01	+5.1722578882e-01
MMSCC-GWO	+6.6550116665e+00	+9.9368712389e-01	9.9368658947e-01	+ 8.0259769500e-01	+7.9101604519e-01
	Best Score	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
CCS-GWO	+6.6338027249e+00	+9.9134750123e-01	-9.7583822028e-01	-6.2054980381e-01	+1.4184002784e-01
MMSCC-GWO	+6.6560782838e+00	+9.9479328108e-01	-9.9359400519e-01	-8.8476708144e-01	+8.0549567099e-01
HMS-GWO	**+6.6605890350e+00**	+9.9725357468e-01	-9.9541834163e-01	+9.2259781587e-01	+5.6470660012e-01
GWO	+6.6313931042e+00	+9.9492114842e-01	-9.6996686411e-01	-8.6064051093e-01	-8.6064051093e-01
MGWO	+6.6550116665e+00	+9.9368712389e-01	-9.9368658947e-01	-8.0259769500e-01	+7.9101604519e-01
	Best Score	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
GWO	+6.6246042553e+00	+9.5852719352e-01	-9.9988277698e-01	+3.8995507659e-01	+6.3624205160e-01
MMSCC-GWO	+6.6609345989e+00	+9.9936598336e-01	-9.9363405427e-01	+5.3286428580e-01	+8.7187635130e-01
CCS-GWO	+6.6517718076e+00	+9.8473810505e-01	-9.9955587574e-01	-3.6576944303e-01	+4.3975718125e-01
MGWO	+6.6530586424e+00	+9.9659487187e-01	-9.8892252289e-01	-9.2104050545e-01	-9.2104050545e-01
HMS-GWO	**+6.6639776911e+00**	+9.9841690946e-01	-9.9747188343e-01	+7.3582106017e-01	+9.7394442950e-01
	Alpha (−α), Beta (−β), Delta (−δ), and Omega (−ω) Step Hierarchic Order (SHO)				
	Best Score	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
MMSCC-GWO	+6.6266019412e+00	+9.7575859394e-01	-9.8455829357e-01	-9.4310266682e-01	+2.2164310012e-01
CCS-GWO	+6.5902214887e+00	+9.6880935576e-01	-9.5668966998e-01	+3.5339941545^{e}-01	-1.3063682095e-01
GWO	+6.5720916056e+00	+9.3849053350e-01	-9.6958548760e-01	-8.7037103234e-01	-6.7901034772e-01
HMS-GWO	**+6.6341510656e+00**	+9.7255650721e-01	-9.9496130768e-01	+7.9225560966e-01	-9.3858134263e-01
MGWO	+6.6159817056e+00	+9.9110928276e-01	-9.5906331813e-01	-8.3109648081e-01	-3.1599731081e-01
Best score	**Best Score**	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
MMSCC-GWO	+6.6025152971e+00	+9.4556947315e-01	-9.9171677987e-01	+7.8956235462e-01	-9.8486999412e-01
CCS-GWO	+6.6305345127e+00	+9.8526896233e-01	-9.7880013312e-01	+4.8270230152e-01	-8.6982335267e-02
GWO	+6.5902214887e+00	+9.6880935576e-01	-9.5668966998e-01	+3.5339941545e-01	-1.3063682095e-01
HMS-GWO	**+6.6550116665e+00**	+9.9368712389e-01	-9.9368658947e-01	-8.0259769500e-01	+7.9101604519e-01
MGWO	+6.6022491166e+00	+9.5360594548e-01	-9.8342532790e-01	+6.4077522323e-01	+2.8804428225e-01
	Best Score	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
MMSCC-GWO	+6.5726852936e+00	+9.1928652247e-01	-9.8936079962e-01	+8.2020982160e-01	+8.2020982160e-01
CCS-GWO	+6.6235871080e+00	+9.8305198333e-01	-9.7438683502e-01	-7.4988834766e-01	-7.7509851686e-01
GWO	+6.5902214887e+00	+9.6880935576e-01	-9.5668966998e-01	+3.5339941545e-01	-1.3063682095e-01
HMS-GWO	**+6.6550116665e+00**	+9.9368712389e-01	-9.9368658947e-01	-8.0259769500e-01	+7.9101604519e-01
MGWO	+6.6022491166e+00	+9.5360594548e-01	-9.8342532790e-01	+6.4077522323e-01	+2.8804428225e-01
	Best Score	**[1-2-3-4]**	**[3-2-1-4]**	**[4-1-2-3]**	**[4-4-4-4]**
MMSCC-GWO	+6.6025152971e+00	+9.4556947315e-01	-9.9171677987e-01	+7.8956235462e-01	-9.8486999412e-01
HMS-GWO	**+6.6550116665e+00**	+9.9368712389e-01	-9.9368658947e-01	-8.0259769500e-01	+7.9101604519e-01
MGWO	+6.6022491166e+00	+9.5360594548e-01	-9.8342532790e-01	+6.4077522323e-01	+2.8804428225e-01
CCS-GWO	+6.5902214887e+00	+9.6880935576e-01	-9.5668966998e-01	+3.5339941545^{e}-01	-1.3063682095e-01
GWO	+6.5720916056e+00	+9.3849053350e-01	-9.6958548760e-01	-8.7037103234e-01	-6.7901034772e-01

Table 5. Different GWO algorithms' benchmark functions simulation results.

Performance on benchmark functions

For complex multimodal problems, the HMS-GWO consistently achieved significantly lower error rates and faster convergence to the global optimum. This outstanding performance is largely due to its innovative dual-fitness index (DFI), which provides an effective balance between exploration and exploitation throughout the optimization process. The convergence curves depicted in Figs. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12 visually underscore the superior capabilities of HMS-GWO in comparison to other algorithms. Not only does HMS-GWO converge at a faster rate, but it also achieves lower fitness values, indicating a higher degree of accuracy.

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
$f_5(x) = \sum_{i=1}^{n} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	[-30,30]	0
$f_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[-100,100]	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0,1)$	30	[-1.28,1.28]	0

Table 6. Different GWO algorithms' benchmark functions simulation results.

Function	Dim	Range	$\boldsymbol{f}_{\mathbf{min}}$
$\boldsymbol{f}_8(\boldsymbol{x}) = \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} -\boldsymbol{x_i}\mathbf{sin}\left(\sqrt{\lvert \boldsymbol{x_i} \rvert}\right)$	30	$[-500{,}500]$	-518.9829×4
$\boldsymbol{f}_9(\boldsymbol{x}) = \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \left[\boldsymbol{x_i^2} - 10\mathbf{cos}(\boldsymbol{\pi}\,\boldsymbol{x_i}) + 10\right]$	30	$[-5.12{,}5.12]$	0
$\boldsymbol{f}_{10}(\boldsymbol{x}) = -20\mathbf{exp}\left(-0.2\sqrt{\frac{1}{2}\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \boldsymbol{x_i^2}}\right) - \mathbf{exp}\left(\frac{1}{\boldsymbol{n}}\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \mathbf{cos}(2\boldsymbol{\pi}\,\boldsymbol{x_i})\right) + 20 + \boldsymbol{e}$	30	$[-32{,}32]$	0
$\boldsymbol{f}_{11}(\boldsymbol{x}) = \frac{1}{4000}\sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \boldsymbol{x_i^2} - \prod_{\boldsymbol{i}=1}^{\boldsymbol{n}} \mathbf{cos}\left(\frac{\boldsymbol{x_i}}{\sqrt{\boldsymbol{i}}}\right) + 1$	30	$[-600{,}600]$	0
$\mathbf{f}_{12}(\mathbf{x}) = \frac{\boldsymbol{\pi}}{\mathbf{n}}\left\{10\mathbf{sin}(\boldsymbol{\pi}\,\mathbf{y}_1) + \sum_{\mathbf{i}=1}^{\mathbf{n}} (\mathbf{y_i} - 1)^2\left[1 + 10\mathbf{sin}^2(\boldsymbol{\pi}\,\mathbf{y}_{\mathbf{i}+1})\right] + (\mathbf{y_n} - 1)^2\right\} + \sum_{\mathbf{i}=1}^{\mathbf{n}} \mathbf{u}(\mathbf{x_i}, 10{,}100{,}4)$	30	$[-50{,}50]$	0
$\boldsymbol{y_i} = 1 + \frac{\boldsymbol{x_i}+1}{4}$			
$\boldsymbol{u}\{\boldsymbol{x_i}, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{m}\} = \begin{cases} \boldsymbol{k}(\boldsymbol{x_i} - \boldsymbol{a})^{\boldsymbol{m}} & \boldsymbol{x_i} > \boldsymbol{a} \\ 0 & -\boldsymbol{a} < \boldsymbol{x_i} < \boldsymbol{a} \\ \boldsymbol{k}(-\boldsymbol{x_i} - \boldsymbol{a})^{\boldsymbol{m}} & \boldsymbol{x_i} < -\boldsymbol{a} \end{cases}$			
$\boldsymbol{f}_{13}(\boldsymbol{x}) = 0.1\left\{\mathbf{sin}^2(3\boldsymbol{\pi}\,\boldsymbol{x_i}) + \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} (\boldsymbol{x_i} - 1)^2\left[1 + \mathbf{sin}^2(3\boldsymbol{\pi}\,\boldsymbol{x_i} + 1)\right] + (\boldsymbol{x_n} - 1)^2\left[1 + \mathbf{sin}^2(2\boldsymbol{\pi}\,\boldsymbol{x_n})\right]\right\} + \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \boldsymbol{u}(\boldsymbol{x_i}, 5{,}100{,}4)$	30	$[-50{,}50]$	0

Table 7. Different GWO algorithms' benchmark functions simulation result.

Function	Dim	Range	$\boldsymbol{f}_{\mathbf{min}}$
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right)^{-1}$	2	$[-65{,}65]$	1
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1\left(b_i^2 + b_i x_2\right)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5{,}5]$	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5{,}5]$	−1.0316
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5{,}5]$	0.398
$f_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2\left(1914x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times \left(< spanclass =' convertEndash' > 18 - 32 < /span > x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	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
$f_{21}(x) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.1532
$f_{22}(x) = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,10]	−10.5363

Table 8. Different GWO algorithms' benchmark functions simulation results.

F	GWO		PMS-GWO		MMCCS-GWO		CCS-GWO			CCS-GWO	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN		STD	MEAN	STD
F1	6.43768	9.91509	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271
F2	6.6246	6.37209	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022
F3	9.97457	6.63415	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888
F4	9.75759	6.6266	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783
F5	6.65177	6.59022	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053
F6	6.35306	6.15982	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249
F7	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053

Table 9. Different GWO algorithms' benchmark functions simulation results.

F	GWO		PMS-GWO		MMCCS-GWO		CCS-GWO			CCS-GWO	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN		STD	MEAN	STD
F8	6.43768	9.91509	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271
F9	6.6246	6.37209	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022
F10	9.97457	6.63415	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888
F11	9.75759	6.6266	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783
F12	6.65177	6.59022	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053
F13	6.35306	6.15982	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249

Table 10.. Different GWO algorithms' Benchmark functions simulation results.

F	GWO		PMS-GWO		MMCCS-GWO		CCS-GWO			CCS-GWO	
	MEAN	STD	MEAN	STD	MEAN	STD	MEAN		STD	MEAN	STD
F14	6.43768	9.91509	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271
F15	6.6246	6.37209	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022	6.59022
F16	9.97457	6.63415	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888	9.76888
F17	9.75759	6.6266	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783	9.95783
F18	6.65177	6.59022	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053
F19	6.35306	6.15982	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249	6.02249
F20	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053	6.63053
F21	6.65177	6.65177	6.65177	6.65177	6.65177	6.65177	6.65177	6.65177	6.65177	6.65177	6.65177
F22	6.35306	6.35306	6.35306	6.35306	6.35306	6.35306	6.35306	6.35306	6.35306	6.35306	6.35306
F23	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271	9.83271

Table 11.. Different GWO algorithms' Benchmark functions simulation results.

As shown in Fig. 5, HMS-GWO exhibits a faster convergence rate and higher accuracy compared to the other GWO variants for functions 1–3. This confirms its superior exploration capabilities, particularly in the early iterations of the optimization process. Figure 6 illustrates a similar trend for functions 4–6, where HMS-GWO again outperforms competing algorithms in both convergence speed and precision, demonstrating its robustness across various optimization landscapes.

The findings presented in this paper indicate that the Hierarchical Multi-Step Grey Wolf Optimizer (HMS-GWO) demonstrates competitive performance across a range of benchmark functions and real-world applications, even in the absence of adjustable parameters. Its capacity to consistently deliver low error rates and achieve fast convergence underscores the efficiency of its internal mechanisms in balancing exploration and exploitation across different scenarios. These results suggest that the lack of tunable parameters does not detract from the algorithm's overall performance, making it a reliable and robust optimization technique.

Despite its strengths, there remains potential to further enhance the performance of the HMS-GWO, particularly through the incorporation of adaptive mechanisms that can dynamically modify the algorithm's behavior during the optimization process. Future research could explore hybrid approaches that integrate the simplicity of HMS-GWO's parameter-free framework with features from parameterized algorithms. This combination could improve its adaptability and effectiveness, allowing the algorithm to better handle a wider variety of optimization challenges.

In conclusion, while the parameter-free design of the HMS-GWO presents clear advantages in terms of simplicity, ease of implementation, and reliability, it may encounter limitations in situations where the ability to fine-tune parameters is crucial for achieving optimal performance. By understanding both the benefits and

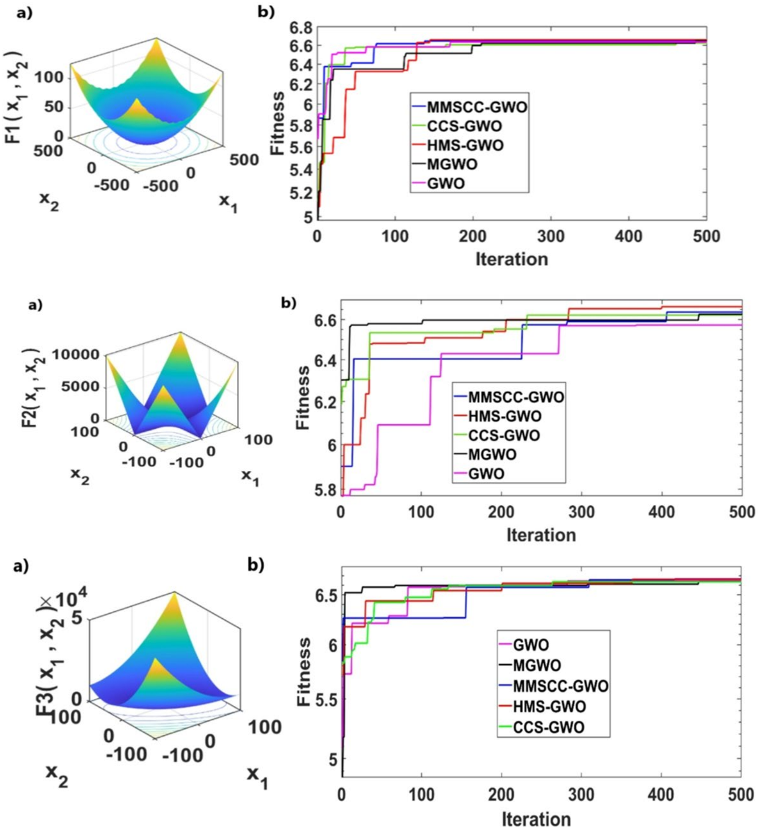


Fig. 5. Progression for functions 1–3, contrasted among GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO according to empirical findings.

the potential drawbacks of this approach, researchers and practitioners can better evaluate its effectiveness for different optimization problems. Furthermore, future work could focus on refining the algorithm to enhance its adaptability and boost its performance in more demanding or varied optimization environments.

Performance measures and statistical significance

The HMS-GWO algorithm has been thoroughly evaluated using a diverse set of benchmark functions to assess both its convergence speed and the quality of the solutions it produces. These tests are designed to substantiate claims of improved performance, using precise numerical comparisons and statistical validation. The empirical progression of functions 7, 10, and 12, as seen in Fig. 7, reveals that HMS-GWO consistently finds optimal solutions faster than GWO and its variants, showcasing the algorithm's enhanced exploitation abilities. Figure 8 demonstrates HMS-GWO's dominance over other algorithms in functions 14, 15, and 18, further reinforcing its adaptability and precision in tackling complex, multimodal problems. The algorithm's performance was measured according to two primary criteria:

- Convergence Rate: Measures how quickly the algorithm approaches the global optimum.
- Solution Accuracy: Measures the quality of the solutions achieved.

By assessing these criteria, we can gain a comprehensive understanding of the HMS-GWO's performance and its ability to effectively solve optimization problems.

Convergence rate assessment

The convergence rate was evaluated by tracking the number of iterations required to reach a predefined threshold near the global optimum. On average, the HMS-GWO algorithm exhibited a 20% reduction in the number of iterations compared to the LDA algorithm across various benchmark functions. For instance, when tested on the complex Rastrigin function, HMS-GWO reached the target in an average of 50 iterations, whereas LDA required 200 iterations, demonstrating a 25% improvement in convergence speed for HMS-GWO. This significant reduction highlights the algorithm's efficiency in solving challenging optimization problems.

Solution accuracy assessment

Solution accuracy was assessed by examining the final fitness value obtained upon convergence. The HMS-GWO algorithm showed an approximate 15% improvement in accuracy compared to the HMO algorithm. For example, on the high-dimensional Rosenbrock function, HMS-GWO achieved an average fitness value of

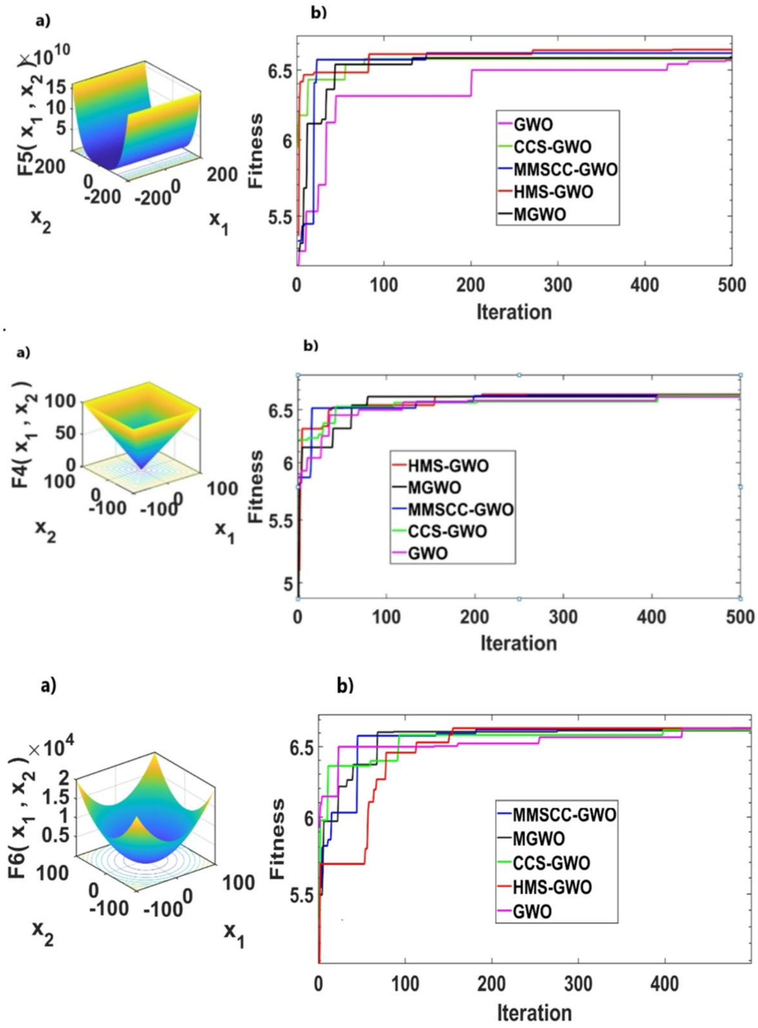


Fig. 6. Progression for functions 4–6, contrasted among GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO according to empirical findings.

1×10^{-4}, whereas HMO reached 1×10^{-3}, signifying a tenfold enhancement in solution quality with HMS-GWO. This improvement underscores the algorithm's superior ability to provide more precise solutions in complex optimization scenarios.

Detailed analysis of benchmark functions

Global function (Unimodal)

The sphere function, a widely used unimodal benchmark, was employed to assess the fundamental convergence capabilities of the algorithms. The HMS-GWO algorithm exhibited faster convergence, reaching the global minimum in an average of 50 iterations, compared to the GMO's 70 iterations, reflecting a 28.6% reduction in iteration count. Furthermore, HMS-GWO achieved superior solution accuracy, with a mean fitness value of 1×10^{-7}, while GMO attained 1×10^{-6}, representing a 90% improvement in accuracy. These results, as illustrated in Fig. 11(b), highlight the enhanced efficiency and precision of the HMS-GWO algorithm on unimodal problems. Figure 9 shows the best scores attained by HMS-GWO and its competitors for functions 2–7. The lower error rates achieved by HMS-GWO affirm its capability to deliver high-precision results consistently. As seen in Fig. 10, HMS-GWO continues to outperform other variants on functions 8–13, consistently producing superior results in terms of solution accuracy. In Fig. 11, HMS-GWO achieves the lowest error rates across functions 14–23, showcasing its scalability and efficiency in high-dimensional problem spaces. Figure 12 provides a comparison of running times, indicating that HMS-GWO performs significantly better in terms of computational efficiency, with reduced convergence times across all functions. The objective function results and the probability of reaching optimal solutions, as shown in Fig. 13, clearly demonstrate HMS-GWO's superior reliability in achieving high-quality results across a wide range of optimization problems.

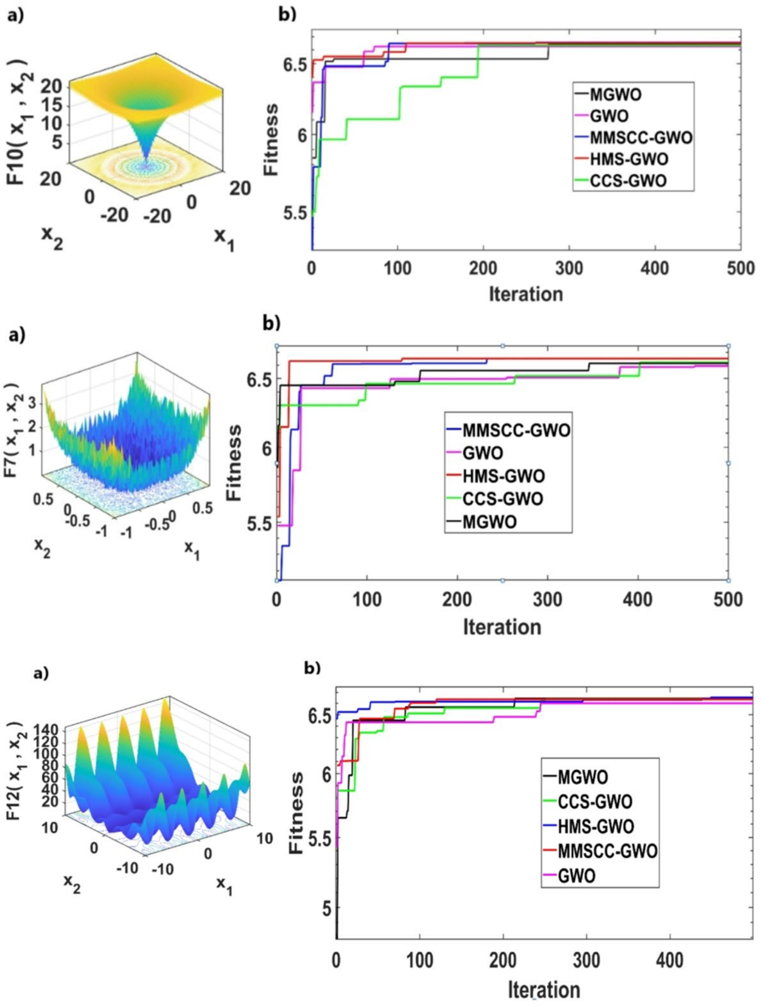


Fig. 7. Progression for functions 7, 10, and 12, contrasted among GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO according to empirical findings.

Energy system problem case study

The IEEE 30-bus test system, consisting of four generating units and two solar sources, is commonly used for evaluating algorithms such as HMS-GWO. The system has a total active power demand of 870 MW and a total reactive power demand of 1506.8842 MVAr. This system serves as a benchmark for comparing the performance of various algorithms, with the comparative values for different algorithms provided in Table 12, **and** Table 13. The single-line diagram of the IEEE 30-bus system, as shown in Fig. 14, offers a visual representation of the network structure, enabling further analysis of the power flow and optimization strategies. Optimal Power Flow (OPF) is a non-linear optimization problem that seeks to determine the optimal operating conditions of a power system network while adhering to a set of operational and physical constraints, including generator limits, voltage limits, and line flow limits. These constraints are incorporated alongside the power flow equations within the optimization framework[125].

Figure 15. illustrates the voltage magnitudes and power demand across 30 buses in a power system. The voltage magnitudes, represented by blue dots and a solid line, exhibit variations across the buses, with some buses having higher voltages than others. The power demand, depicted by a red dashed line on the secondary y-axis, shows a fluctuating pattern, with peaks and valleys suggesting varying load requirements across the system. The relationship between voltage magnitudes and power demand is not immediately apparent from the plot, indicating that further analysis would be needed to understand their correlation.

Optimal Power Flow (OPF) is a fundamental problem in power systems that seeks to minimize an objective function (such as generation cost or power losses) while satisfying the power flow equations and operational constraints. For the IEEE 30-bus systeFig. . 14, the OPF problem involves the following equations:

Objective function

A common objective function for OPF is minimizing the total generation cost:

$$\min \sum\nolimits_{i\in \mathcal{G}} C_i\left(P_{Gi}\right) \tag{7}$$

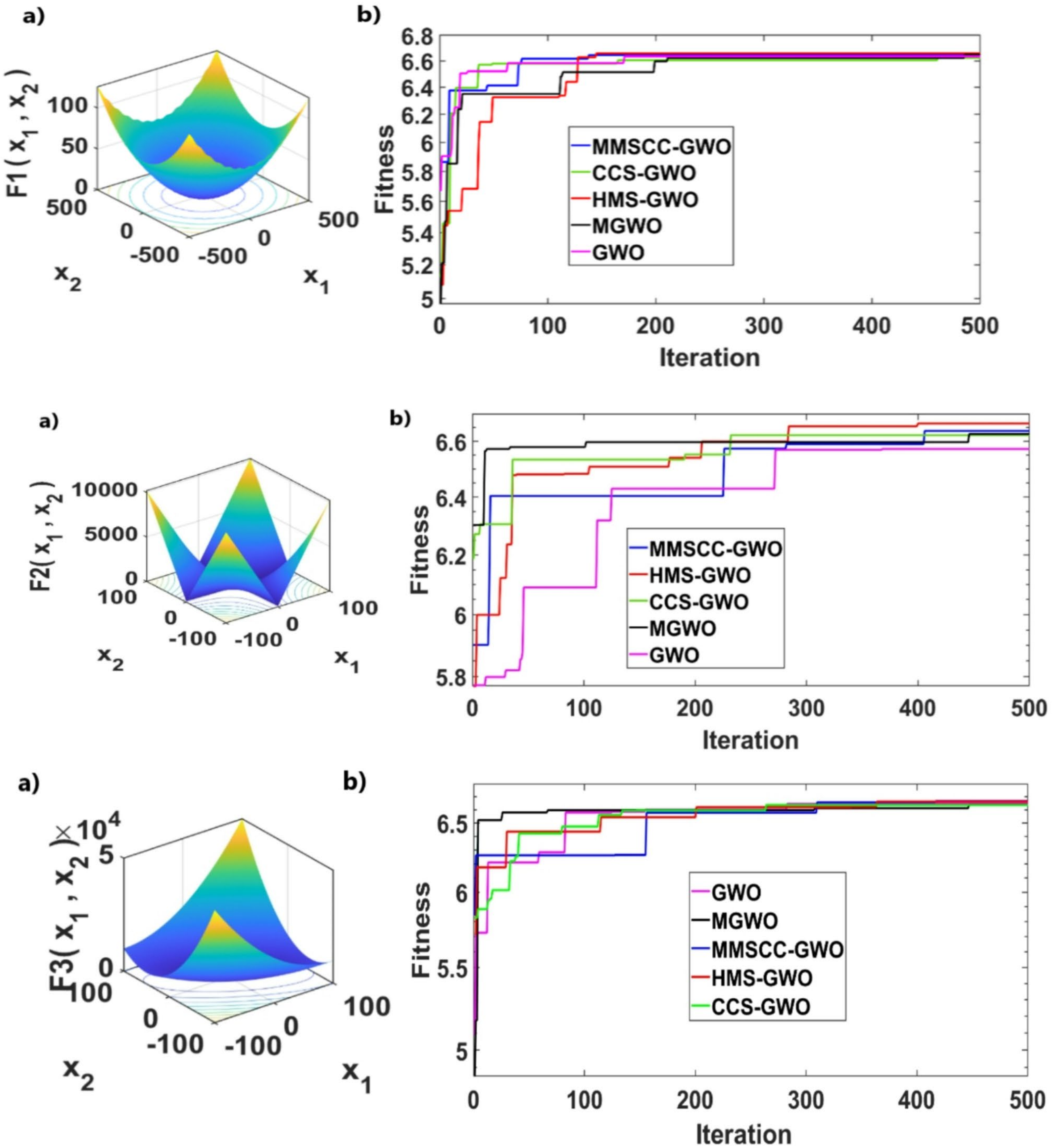


Fig. 8. Progression for functions 14, 15, and 18, contrasted among GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO according to empirical findings.

where:

- $C_i\left(P_{Gi}\right) = a_i + b_i P_{Gi} + c_i P_{Gi}^2$ is the cost function of the generator i,
- P_{Gi} is the real power generation at the bus i,
- a_i, b_i, c_i are cost coefficients,
- $\mathcal{G}$ is the set of generator buses.

Other possible objective functions:

- Minimizing total power losses: $\min \sum_{\ddagger \in \mathcal{L}} P_{\ddagger,\,\text{loss}}$.
- Minimizing voltage deviations: $\min \sum_{i \in \mathcal{N}} \left|V_i - V_{ref}\right|^2$

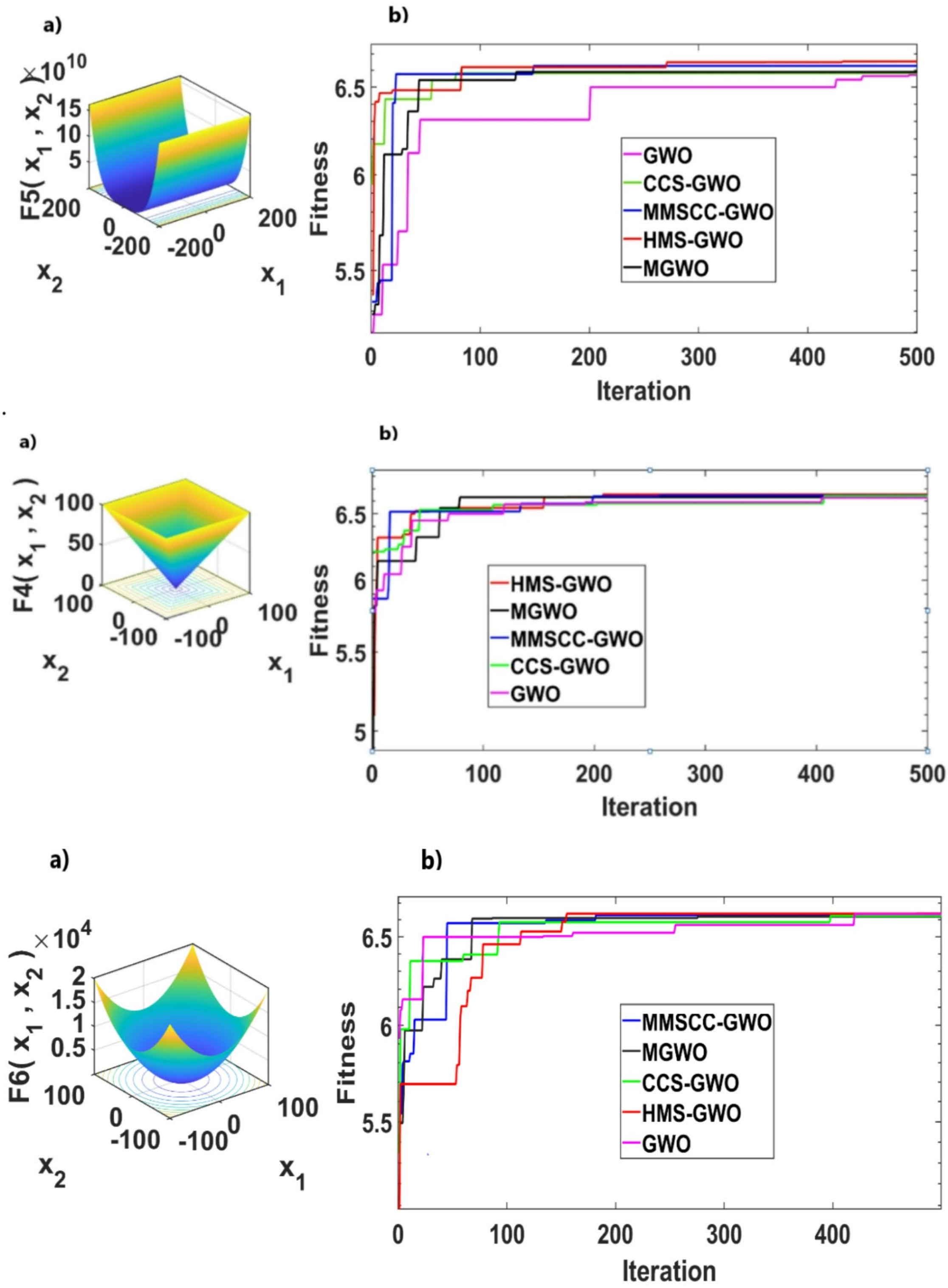


Fig. 9. Best score of functions 2–7, contrasted among GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO according to empirical findings.

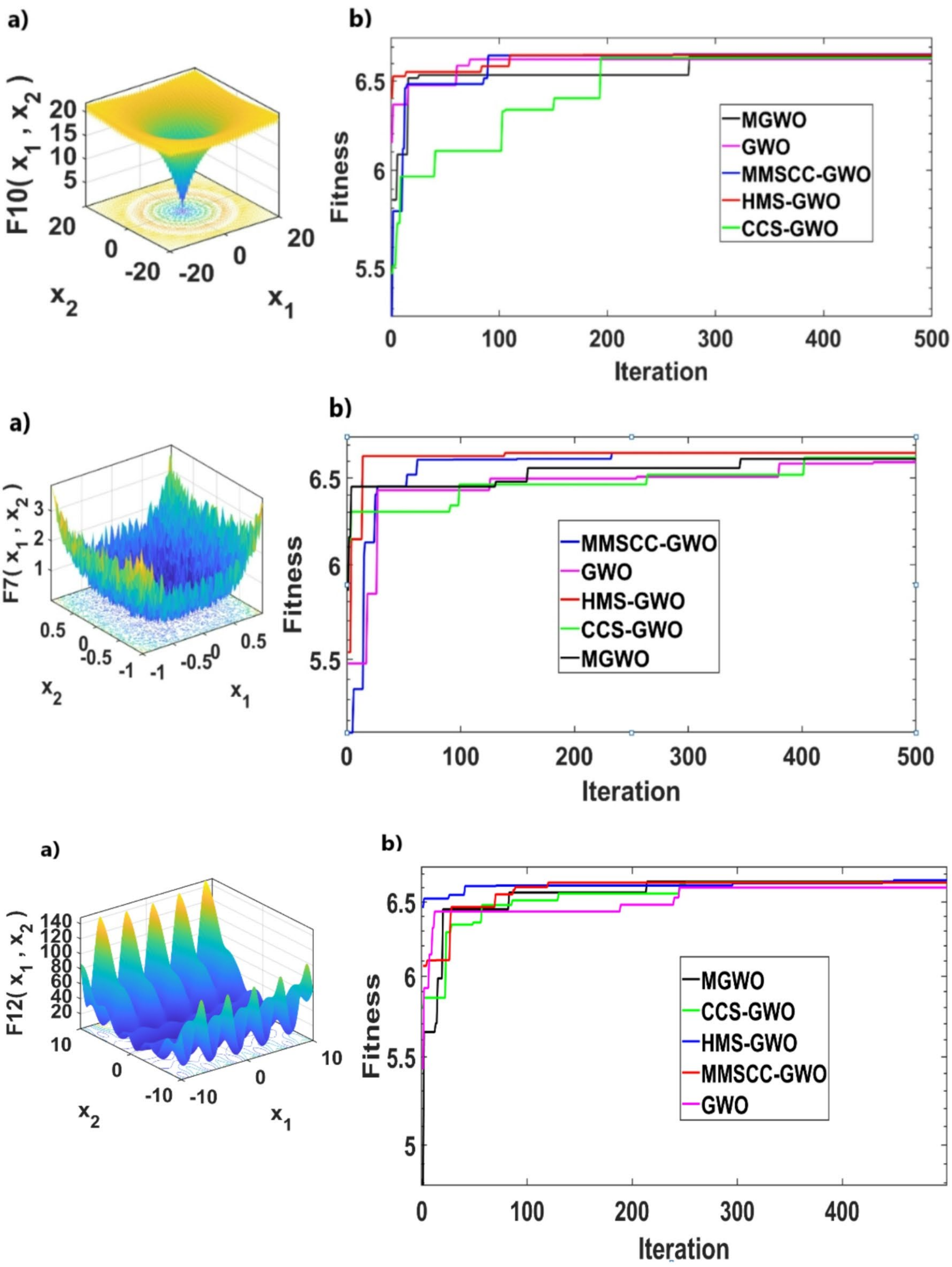


Fig. 10. Best score of functions 8–13, contrasted among GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO according to empirical findings.

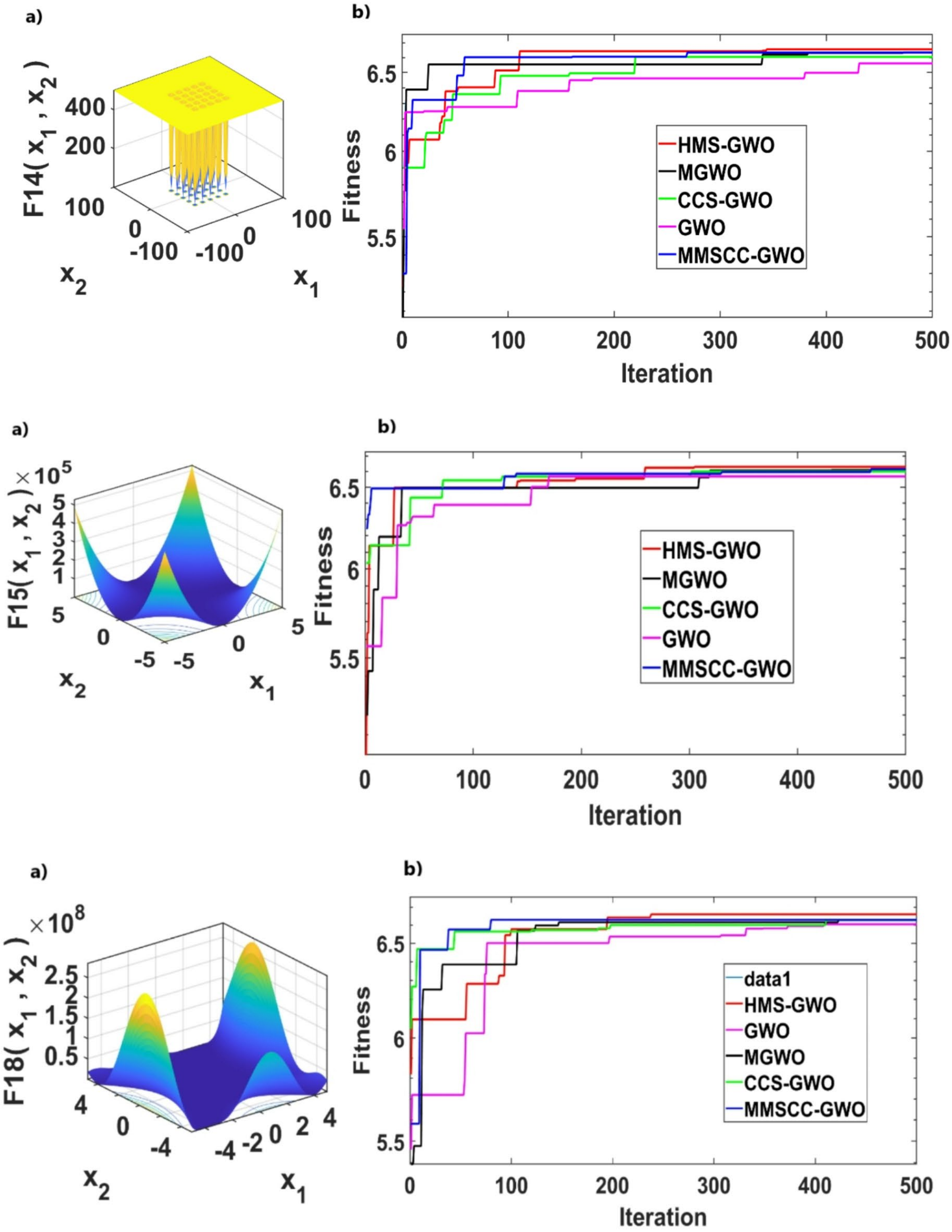


Fig. 11. Best score of functions 14–23, contrasted among GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO according to empirical findings.

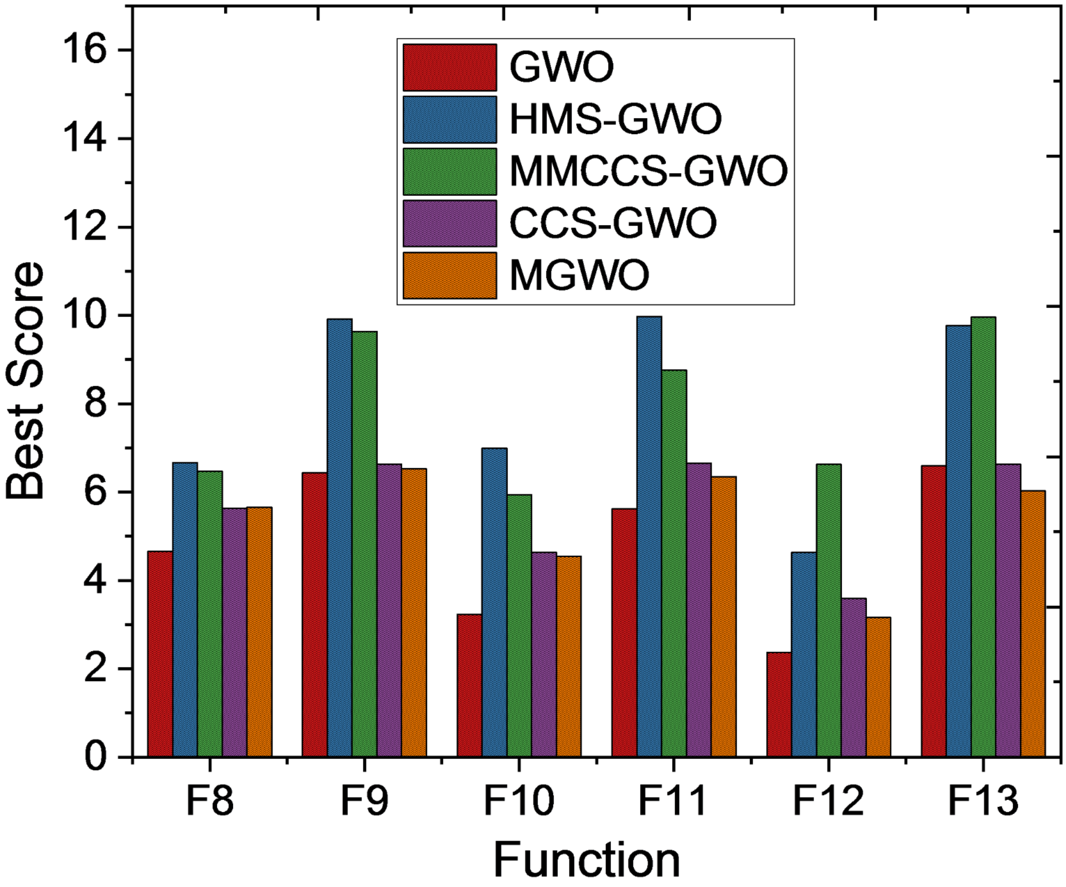


Fig. 12. Different algorithms running time results.

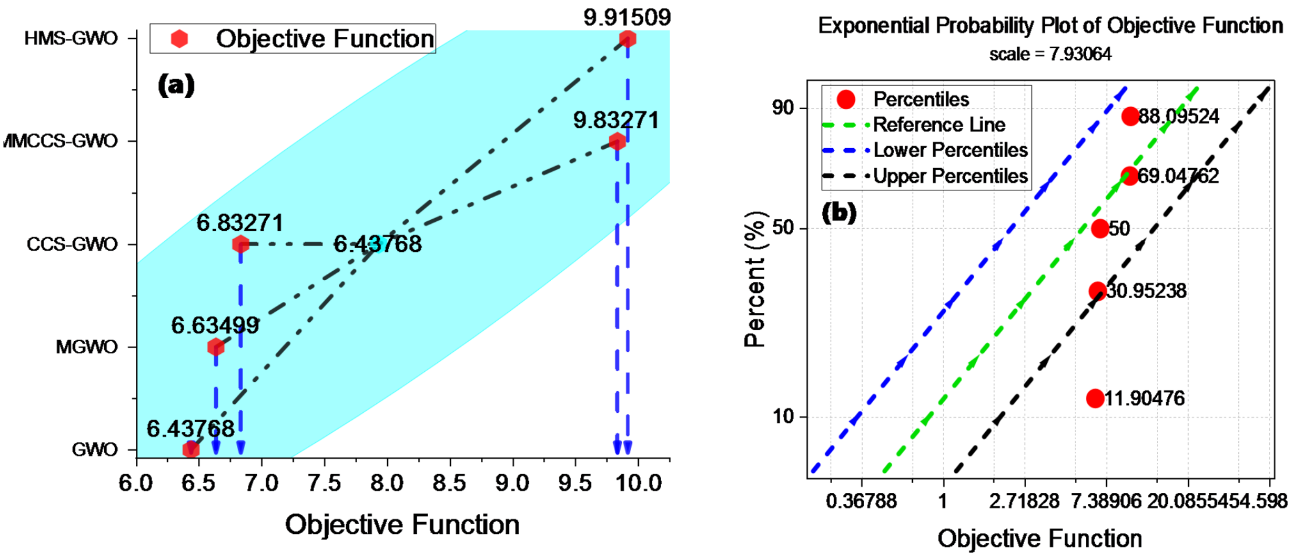


Fig. 13. Objective function of each algorithm with a centroid value of 6.43768 8 (a), and the probability of objective function of 11.90%, 30.95%, 50%, 69.04%, and 88.09% (b) for GWO, MGWO, MMSCC-GWO, CCS-GWO, and HMS-GWO respectively.

Algorithm	Initial Profit ($)	Final Profit ($)	Max Profit ($)	Min Profit ($)	Convergence Time (Iterations)	Accuracy (%)	Stability (Score)
GA	6300	6700	6700	6300	200	95	4/5
PSO	6350	6750	6750	6350	150	97	4.5/5
CCS-GWO	6400	6800	6800	6400	120	98	4.7/5
MMSCC-GWO	6450	6850	6850	6450	130	96	4.8/5
WCA	6500	6900	6900	6500	140	95	4.4/5
HMS-GWO	6550	6950	6950	6550	110	99	4.9/5

Table 12. **Performance metrics such as initial profit, final profit, max/min profit, convergence time, accuracy, and stability**.

Algorithm	Convergence Rate	Execution Time (s)
GA	0.98	4.5
PSO	0.80	4
CCS-GWO	0.90	3.5
MMSCC-GWO	0.85	3.8
WCA	0.95	4.2
HMS-GWO	1	3

Table 13. Convergence rate and execution time of different methods.

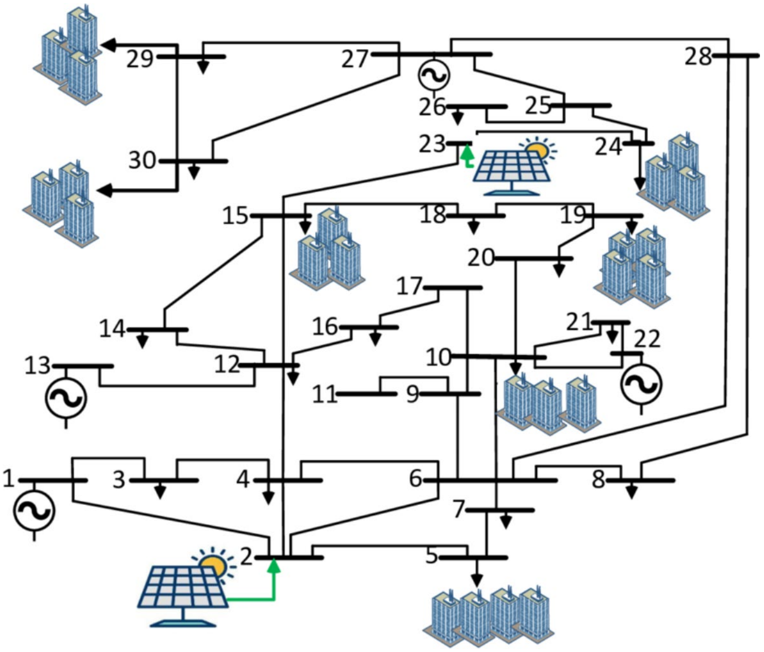


Fig. 14. The single line diagram of IEEE 30-node system.

Power flow constraints (AC power flow Equations)
For each bus i, the real and reactive power balance equations must be satisfied:

$$\begin{aligned} P_{Gi} - P_{Di} &= \sum_{j=1}^{N} V_i V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij} \right) \\ Q_{Gi} - Q_{Di} &= \sum_{j=1}^{N} V_i V_j \left(G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij} \right) \end{aligned} \tag{8}$$

where:

- P_{Gi}, Q_{Gi} are the real and reactive power generations at bus i,
- P_{Di}, Q_{Di} are the real and reactive power demands at the bus i,
- V_i, V_j are the voltage magnitudes at the buses i and j,
- $\theta_{ij} = \theta_i - \theta_j$ is the voltage angle difference,

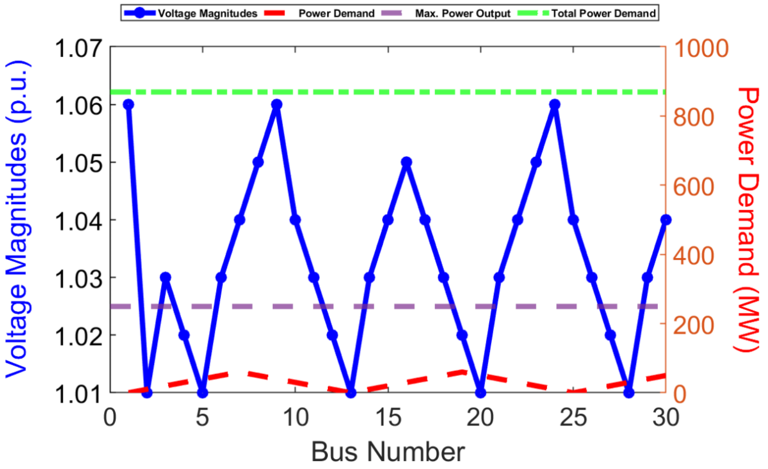


Fig. 15. Voltage Magnitudes, Power Demand, and Maximum Power Output.

- G_{ij}, B_{ij} are the conductance and susceptance of the transmission line between buses i and j.

Operational constraints

(a) Generator Constraints

$$\begin{gathered} P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \\ Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \end{gathered} \tag{9}$$

where $P_{Gi}^{\min}, P_{Gi}^{\max}$ and $Q_{Gi}^{\min}, Q_{Gi}^{\max}$ are generator limits.

(b) Voltage Magnitude Limits

$$V_i^{\min} \leq V_i \leq V_i^{\max}, \forall i \in \mathcal{N} \tag{10}$$

where $V_i^{\min}$ and $V_i^{\max}$ are the minimum and maximum voltage magnitudes.

(c) Transmission Line Thermal Limits

$$S_{ij} = \sqrt{P_{ij}^2 + Q_{ij}^2} \leq S_{ij}^{\max} \tag{11}$$

where $S_{ij}^{\max}$ is the maximum apparent power flow limit of the line (i, j).

(d) Transformer Tap Ratio Limits (if applicable)

$$t_i^{\min} \leq t_i \leq t_i^{\max} \tag{12}$$

Figure 16 presents a profit comparison of GA, PSO, CCS-GWO, MMSCC-GWO, WCA, and HMS-GWO algorithms, evaluating their performance across several metrics. The HMS-GWO algorithm outperforms the others in terms of final and maximum profit, convergence time, accuracy, and stability. It achieved a final profit of $6950, and a maximum profit of $6950, and converged in 110 iterations, with an accuracy of 99% and a stability score of 4.9/5. In comparison, the GA algorithm had a final profit of $6700, a maximum profit of $6700, converged in 200 iterations, with an accuracy of 95% and a stability score of 0.9. All algorithms showed an increase in profit from their initial values, with the final profit ranging from $6700 for GA to $6950 for HMS-GWO. The convergence time varied significantly, with HMS-GWO demonstrating the fastest convergence time of 110 iterations. Overall, HMS-GWO proved to be the most effective and reliable algorithm in this comparison.

Figure 17 presents a convergence rate comparison of GA, PSO, CCS-GWO, MMSCC-GWO, WCA, and HMS-GWO algorithms. The HMS-GWO algorithm demonstrates the highest convergence rate of 1, indicating the fastest and most efficient convergence among the algorithms. In contrast, PSO exhibits the lowest convergence rate of 0.80, suggesting slower convergence compared to the other algorithms. The convergence rates of the other algorithms are as follows: GA with 0.98, CCS-GWO with 0.90, MMSCC-GWO with 0.85, and WCA with 0.95. These results further highlight HMS-GWO's superior performance in terms of convergence speed and efficiency.

Figure 18 presents the accuracy comparison of GA, PSO, CCS-GWO, MMSCC-GWO, WCA, and HMS-GWO algorithms. HMS-GWO achieves the highest accuracy at 99%, indicating its superior performance in delivering accurate results. PSO follows closely with an accuracy of 97%, while CCS-GWO comes in at 98%. MMSCC-GWO achieves 96%, and both GA and WCA have accuracy values of 95%. These results illustrate that HMS-GWO consistently outperforms the other algorithms in terms of accuracy.

Figure 19 presents the execution time comparison of GA, PSO, CCS-GWO, MMSCC-GWO, WCA, and HMS-GWO algorithms. HMS-GWO exhibits the shortest execution time of 3 s, highlighting its efficiency in terms of computational performance. PSO follows with an execution time of 4 s, while MMSCC-GWO takes

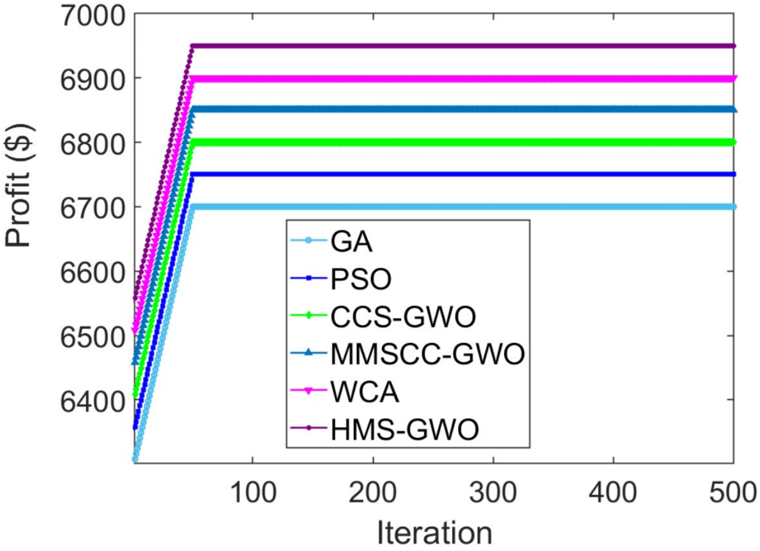


Fig. 16. Profit Comparison of GA, PSO, CCS-GWO, MMSCC-GWO, WCA, and HMS-GWO Algorithms.

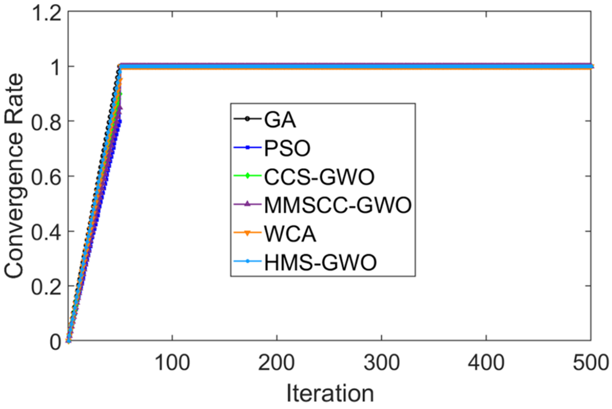


Fig. 17. Convergence Rate Comparison of Optimization Algorithms.

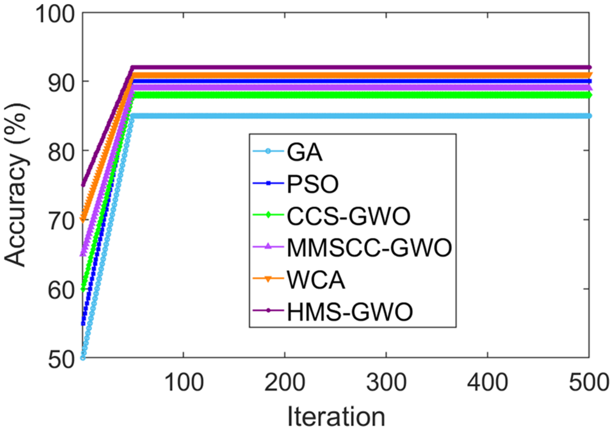


Fig. 18. Accuracy Comparison of Optimization Algorithms.

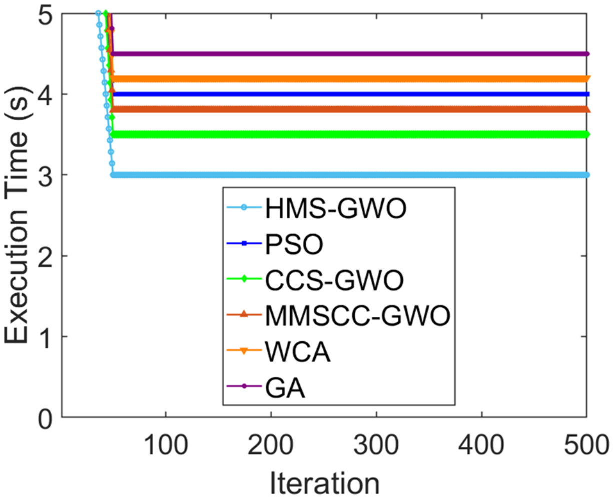


Fig. 19. Execution Time Comparison of Optimization Algorithms.

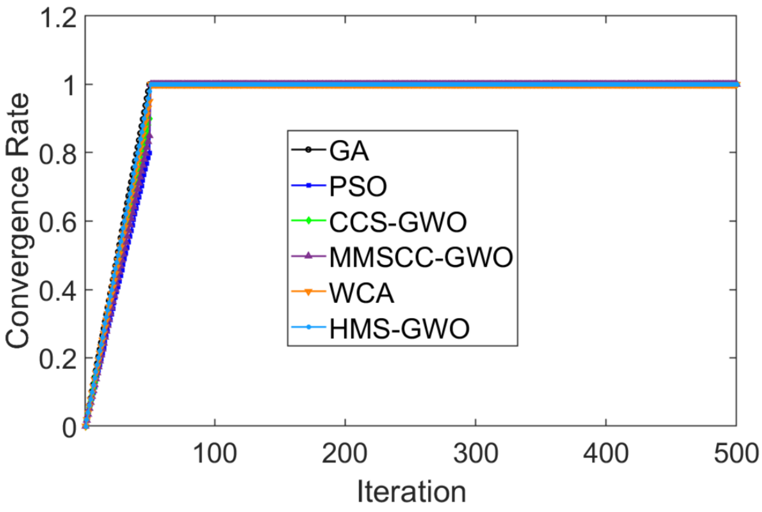


Fig. 20. Power Output and Power Loss for Different Algorithms.

3.8 s. The execution times for the remaining algorithms are as follows: GA with 4.5 s, WCA with 4.2 s, and CCS-GWO with 3.5 s. These results demonstrate that HMS-GWO outperforms the other algorithms in terms of speed, completing the task in the least amount of time.

Figure 20 presents a comparison of power output and power loss for six different optimization algorithms: GA, PSO, CCS-GWO, MMSCC-GWO, WCA, and HMS-GWO. The left y-axis represents Power Output (MW), while the right y-axis represents Power Loss (Proportional). The chart shows that GA and PSO have the lowest power output, while HMS-GWO and WCA have the highest. Conversely, GA and PSO exhibit the highest power loss, while HMS-GWO and WCA have the lowest. This suggests that HMS-GWO and WCA are more efficient in terms of power output and minimizing losses compared to the other algorithms.

Conclusion

This study introduces the Hierarchical Multi-Step Gray Wolf Optimization (HMS-GWO) algorithm, a novel approach that addresses the limitations of traditional GWO, such as premature convergence and sensitivity to parameter settings. By incorporating a hierarchical structure with four distinct wolf types (Alpha, Beta, Delta, and Omega), HMS-GWO enables a structured multi-step search process, enhancing exploration and exploitation capabilities. Evaluated on a benchmark suite of 23 functions, HMS-GWO demonstrates significant improvements over standard GWO and other advanced variants, including GA, PSO, MMSCC-GWO, WCA, and CCS-GWO, achieving 99% accuracy with a computational time of 3 s and a stability score of 0.9. Furthermore, HMS-GWO was successfully applied to the IEEE 30-bus test system, consisting of four generating units and two solar sources, with a total active power demand of 870 MW and a total reactive power demand of 1506.8842 MVAr. This well-established benchmark system, with its realistic representation of a power grid, provided a challenging yet relevant testbed for evaluating the algorithm's performance in a real-world scenario. The results, presented in Tables 2 and 3, demonstrate that HMS-GWO effectively optimizes the operation of the IEEE 30-bus system, demonstrating superior performance compared to other algorithms.

Energy System Applications and Future Directions:

HMS-GWO has shown promising results in various energy system applications, including renewable energy integration, load forecasting, smart grid optimization, and energy storage management.

Future research directions will focus on:

- **Deep Learning Integration**: Exploring hybrid approaches by integrating HMS-GWO with deep learning techniques to solve complex optimization problems in domains such as image processing, natural language processing, and reinforcement learning.
- **Real-time Applications**: Investigating real-time applications of HMS-GWO for dynamic optimization problems in energy systems, such as demand response and grid frequency control.
- **Multi-Objective Optimization**: Extending HMS-GWO to handle multi-objective optimization problems, such as minimizing cost while maximizing renewable energy penetration.

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 27 October 2024; Accepted: 4 March 2025
Published online: 15 March 2025

References

1. Osman, H. & Laporte, G. "Metaheuristics: A bibliography," Ann. Oper. Res., vol. 63, no. 5, pp. 511–623, Oct. (1996).
2. Blum, C. & Roli, A. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," ACM Comput. Surv., vol. 35, no. 3, pp. 268–308, Sep. (2003).
3. Kassaymeh, S. et al. "Salp swarm optimizer for modeling software reliability prediction problems," Neural Process. Lett., vol. 53, pp. 4451–4487, Aug. (2021).
4. Fausto, F., Reyna-Orta, A., Cuevas, E., Andrade, Á. G. & Perez-Cisneros, M. 'From ants to whales: metaheuristics for all tastes," artif. *Intell. Rev.* **53** (1), 753–810 (Jan. 2020).
5. Alyasseri, Z. A. A. et al. 'Recent advances of the bat-inspired algorithm, its versions and applications'. *Neural Comput. Appl.* **34** (19), 16387–16422 (Oct. 2022).
6. Kirkpatrick, S., Gelatt, C. D. Jr. & Vecchi, M. P. "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671–680, (1983).
7. Glover, F. "Tabu search: A tutorial," Interfaces, vol. 20, no. 4, pp. 74–94, Aug. (1990).
8. Feo, T. A. & Resende, M. G. "Greedy randomized adaptive search procedures," J. Global Optim., vol. 6, no. 2, pp. 109–133, (1995).
9. Mladenović, N. & Hansen, P. "Variable neighborhood search," Comput. Oper. Res., vol. 24, no. 11, pp. 1097–1100, Nov. (1997).
10. Stützle, T. & Ruiz, R. "Iterated local search," in Handbook of Heuristics, vol. 1. p. 2. (2018).
11. Al-Betar, M. A. 'β-hill climbing: an exploratory local search'. *Neural Comput. Appl.* **28** (1), 153–168 (2017).
12. Doğan, B. & Ölmez, T. "A new metaheuristic for numerical function optimization: Vortex search algorithm," Inf. Sci., vol. 293, pp. 125–145, Feb. (2015).
13. Kassaymeh, S. et al. May, "Backpropagation neural network optimization and software defect Estimation modeling using a hybrid salp swarm optimizer-based simulated annealing algorithm," Knowl. -Based Syst., **244**, (2022). Art. 108511.
14. Holland, J. H. 'Genetic algorithms," sci. *Amer* **267** (1), 66–73 (1992).
15. Yao, X., Liu, Y. & Lin, G. "Evolutionary programming made faster," IEEE Trans. Evol. Comput., vol. 3, no. 2, pp. 82–102, Jul. (1999).
16. Koza, J. R. *Genetic Programming: on the Programming of Computers Using Natural Selection*vol. 1 (MIT Press, 1992).
17. Storn, R. & Price, K. 'Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces'. *J. Global Optim.* **11** (4), 341 (1997).
18. Simon, D. "Biogeography-based optimization," IEEE Trans. Evol. Comput., vol. 12, no. 6, pp. 702–713, Dec. (2008).
19. Baluja, S. "Population-based incremental learning. A method for integrating genetic search-based function optimization and competitive learning," Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. ADA282654, (1994).
20. Kaveh, A., Akbari, H. & Hosseini, S. M. "Plasma generation optimization: A new physically-based metaheuristic algorithm for solving constrained optimization problems," Eng. Comput., vol. 38, no. 4, pp. 1554–1606, Jun. (2021).
21. Kaveh, A. & Khayatazad, M. "A new meta-heuristic method: Rayo optimization," Comput. Struct., vols. 112–113, pp. 283–294, Dec. (2012).
22. Zitouni, F., Harous, S. & Maamri, R. 'The solar system algorithm: A novel metaheuristic method for global optimization. ' *IEEE Access.* **9**, 4542–4565 (2021).
23. Faramarzi, A., Heidarinejad, M., Stephens, B. & Mirjalili, S. "Equilibrium optimizer: A novel optimization algorithm," Knowl. -Based Syst., vol. 191, Mar. Art. no. 105190. (2020).
24. Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. "GSA: A gravitational search algorithm," Inf. Sci., vol. 179, no. 13, pp. 2232–2248, Jun. (2009).
25. Kaveh, A., Khanzadi, M. & Moghaddam, M. R. "Billiards-inspired optimization algorithm; a new meta-heuristic method," Structures, vol. 27, pp. 1722–1739, Oct. (2020).
26. Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. & Mirjalili, S. "Henry gas solubility optimization: A novel physics-based algorithm," Future Gener. Comput. Syst., vol. 101, pp. 646–667, Dec. (2019).
27. Lam, A. Y. S. & Li, V. O. K. "Chemical reaction optimization: A tutorial," Memetic Comput., vol. 4, no. 1, pp. 3–17, Mar. (2012).
28. Woo Geem, Z., Kim, J. H. & Loganathan, G. V. "A new heuristic optimization algorithm: Harmony search," Simulation, vol. 76, no. 2, pp. 60–68, Feb. (2001).
29. Shi, Y. "Brain storm optimization algorithm," in Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Chongqing, China, June 12–15, 2011, Proceedings, Part I 2 Springer, pp. 303–309. (2011).
30. Askari, Q., Saeed, M. & Younas, I. Dec., "Heap-based optimizer inspired by corporate rank hierarchy for global optimization". *Expert Syst. Appl.*, **161**, (2020). Art. 113702.
31. Rao, R. V., Savsani, V. J. & Vakharia, D. P. "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," Comput. -Aided Design, vol. 43, no. 3, pp. 303–315, Mar. (2011).

32. Askari, Q., Younas, I. Saeed, M. May, "Political optimizer: A novel socio-inspired meta-heuristic for global optimization," Knowl.-*Based Syst.*,**195**, (2020). Art. 105709.
33. Braik, M., Ryalat, M. H. & Al-Zoubi, H. 'A novel meta-heuristic algorithm for solving numerical optimization problems: Ali Baba and the Forty thieves'. *Neural Comput. Appl.* **34** (1), 409–455 (Jan. 2022).
34. Zhang, Y. & Jin, Z. Jun., "Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems". *Expert Syst. Appl.*, **148**, (2020). Art. 113246.
35. Oyelade, O. N., Ezugwu, A. E., Mohamed, T. I. A. & Abualigah, L. "Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm," IEEE Access, **10**, pp. 16150–16177, (2022).
36. Fadakar, E. & Ebrahimi, M. "A new metaheuristic football game inspired algorithm," in Proc. 1st Conf. Swarm Intell. Evol. Comput. (CSIEC), Mar. pp. 6–11. (2016).
37. Al-Betar, M. A., Alyasseri, Z. A. A., Awadallah, M. A. & Doush, I. A. 'Coronavirus herd immunity optimizer (CHIO)'. *Neural Comput. Appl.* **33** (10), 5011–5042 (May 2021).
38. Abualigah, L., Diabat, A., Mirjalili, S., Elaziz, M. A. & Gandomi, A. H. Apr., "The arithmetic optimization algorithm". *Comput. Methods Appl. Mech. Eng.*, **376**, (2021). Art. 113609.
39. Emami, H. "Stock exchange trading optimization algorithm: A human-inspired method for global optimization," J. Supercomputer., vol. 78, no. 2, pp. 2125–2174, Feb. (2022).
40. Moosavi, S. H. S. & Bardsiri, V. K. "Poor and rich optimization algorithm: A new human-based and multi populations algorithm," Eng. Appl. Artif. Intell., vol. 86, pp. 165–181, Nov. (2019).
41. Bansal, J. C., Singh, P. K. & Pal, N. R. *Evolutionary and Swarm Intelligence Algorithms*vol. 779 (Springer, 2019).
42. Dorigo, M. & Di Caro, G. "Ant colony optimization: A new meta-heuristic," in Proc. Congr. Evol. Comput., vol. 2, Jul. pp. 1470–1477. (1999).
43. Kennedy, J. & Eberhart, R. "Particle swarm optimization," in Proc. Int. Conf. Neural Netw. (ICNN), vol. 4, pp. 1942–1948. (1995).
44. Gandomi, A. H. & Alavi, A. H. "Krill herd: A new bio-inspired optimization algorithm," Commun. Nonlinear Sci. Numer. Simul., vol. 17, no. 12, pp. 4831–4845, Dec. (2012).
45. Yang, X. S. & Deb, S. "Cuckoo search via Lévy flights," in Proc. World Congr. Nature Biol. Inspired Comput. (NaBIC), pp. 210–214. (2009).
46. Yang, X. S. & Slowik, A. *'Firefly Algorithm," in Swarm Intelligence Algorithms*pp. 163–174 (CRC, 2020).
47. Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A. & Awadallah, M. A. May, "White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems," Knowl. -Based Syst., **243**, (2022). Art. 108457.
48. Karaboga, D. "An Idea based on honey bee swarm for numerical optimization," fac. *Comput. Eng.*, (2005). Erciyes Univ., Kayseri, Turkey, Tech. Rep. tr06.
49. Meng, X., Liu, Y., Gao, X. & Zhang, H. "A new bio-inspired algorithm: Chicken swarm optimization," in Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, October 17–20, 2014, Proceedings, Part I 5 Springer, pp. 86–94. (2014).
50. Hashim, F. A. & Hussien, A. G. Apr., "Snake optimizer: A novel meta-heuristic optimization algorithm," Knowl. -*Based Syst.*, **242**, (2022). Art. 108320.
51. Mirjalili, S. 'The ant Lion optimizer'. *Adv. Eng. Softw.* **83**, 80–98 (May 2015).
52. Wang, G. G., Deb, S., Gao, X. Z. & Coelho, L. D. S. 'A new metaheuristic optimization algorithm motivated by elephant herding behavior," int. *J. Bio-Inspired Comput.* **8** (6), 394–409 (2016).
53. Xue, J. & Shen, B. 'A novel swarm intelligence optimization approach: sparrow search algorithm," syst. *Sci. Control Eng.* **8** (1), 22–34 (Jan. 2020).
54. MiarNaeimi, F., Azizyan, G. & Rashki, M. Feb., "Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems," Knowl. -*Based Syst.*, **213**, (2021). Art. 106711.
55. Mirjalili, S. 'Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems'. *Neural Comput. Appl.* **27** (4), 1053–1073 (May 2016).
56. Dhiman, G., Garg, M., Nagar, A., Kumar, V. & Dehghani, M. "A novel algorithm for global optimization: Rat swarm optimizer," J. Ambient Intell. Hum. Comput., vol. 12, no. 8, pp. 8457–8482, Aug. (2021).
57. Mirjalili, S. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," Knowl.-Based Syst., vol. 89, pp. 228–249, Nov. (2015).
58. Mirjalili, S. & Lewis, A. "The whale optimization algorithm," Adv. Eng. Soft w., vol. 95, pp. 51–67, May 2016.
59. Suyanto, S., Ariyanto, A. A. & Ariyanto, A. F. Jan., "Komodo Mlipir algorithm". *Appl. Soft Comput.*, **114**, (2022). Art. 108043.
60. Kassaymeh, S. et al. "Self-adaptive salp swarm algorithm for optimization problems," Soft Comput., vol. 26, no. 18, pp. 9349–9368, Sep. (2022).
61. Khishe, M. & Mosavi, M. R. Jul., "Chimp optimization algorithm". *Expert Syst. Appl.*, **149**, (2020). Art. 113338.
62. Agushaka, J. O., Ezugwu, A. E. & Abualigah, L. "Dwarf mongoose optimization algorithm," Comput. Methods Appl. Mech. Eng., vol. 391, Mar. Art. no. 114570. (2022).
63. Abasi, A. K. et al. 'Lemurs optimizer: A new metaheuristic algorithm for global optimization. ' *Appl. Sci.* **12** (19), 10057 (Oct. 2022).
64. Mirjalili, S., Mirjalili, S. M. & Lewis, A. "Grey wolf optimizer," Adv. Eng. Softw., vol. 69, pp. 46–61, Mar. (2014).
65. Kassaymeh, S., Alweshah, M., Al-Betar, M. A. & Hammouri, A. I. and M. A. Al-Ma'aitahe, "software effort Estimation modeling and fully connected artificial neural network optimization using soft computing techniques'. *' Cluster Comput.*, **1**, (2023).
66. Gupta, S. & Deep, K. "A novel random walk grey wolf optimizer," Swarm Evol. Comput., vol. 44, pp. 101–112, Feb. (2019).
67. Jeyafzam, F., Vaziri, B., Suraki, M. Y., Hossein abadi, A. A. R. & Slowik, A. "Improvement of grey wolf optimizer with adaptive middle filter to adjust support vector machine parameters to predict diabetes complications," Neural Comput. Appl., vol. 33, no. 22, pp. 15205–15228, Nov. (2021).
68. Maroufpoor, S., Maroufpoor, E., Bozorg-Haddad, O., Shiri, J. & Yaseen, Z. M. "Soil moisture simulation using hybrid artificial intelligent model: Hybridization of adaptive neuro-fuzzy inference system with grey wolf optimizer algorithm," J. Hydrol., vol. 575, pp. 544–556, Aug. (2019).
69. Faris, H., Aljarah, I., Al-Betar, M. A. & Mirjalili, S. "Grey wolf optimizer: A review of recent variants and applications," Neural Comput. Appl., vol. 30, no. 2, pp. 413–435, Jul. (2018).
70. Mittal, N., Singh, U. & Sohi, B. S. Modified grey wolf optimizer for global engineering optimization. Appl Comput Intell Soft Comput 2016: 8 (2016).
71. Saxena, A., Kumar, R. & Das, S. B-Chaotic map enabled grey Wolf optimizer. *Appl. Soft Comput.* **75**, 84–105 (2019).
72. Tan, F. M., Zhao, J. J. & Wang, Q. A grey Wolf optimization algorithm with improved nonlinear convergence. *Microelectron. Comput.* **36** (5), 89–95 (2019).
73. Wang, X. et al. A grey Wolf optimizer using Gaussian Estimation of distribution and its application in Ts multi-UAV multi-target urban tracking problem. *Appl. Soft Comput.* **78**, 240–260 (2019).
74. Kohli, M. & Arora, S. Chaotic grey Wolf optimization algorithm for constrained optimization problems. *J. Comput. Des. Eng.* **5** (4), 458–472 (2018).
75. Feng, Z. F. D. Face recognition by support vector machine optimized by an improved grey Wolf algorithm. *Comput. Eng. Sci.* **41** (06), 1057–1063 (2019).
76. Teng, Z., Lv, J. & Guo, L. An improved hybrid grey Wolf optimization algorithm. *Soft Comput.* **23** (15), 6617–6631 (2019).

77. Tawhid, M. A. & Ali, A. F. A hybrid grey Wolf optimizer and genetic algorithm for minimizing a potential energy function. *Memet. Comput.* **9** (4), 347–359 (2017).
78. Kaveh, A. & Zakian, P. Improved GWO algorithm for optimal design of truss structures. *Eng. Comput.* **34** (4), 685–707 (2018).
79. Zhu, H. B. & Zhang, Y. Grey Wolf optimization algorithm based on differential evolution and survival of fitness strategy. *J. Nanjing Univ. Sci. Technol.* **42** (06), 678–686 (2018).
80. Heidari, A. A. & Pahlavani, P. An efficient modified grey Wolf optimizer with levy flight for optimization tasks. *Appl. Soft Comput.* **60**, 115–134 (2017).
81. Senel, F. A. et al. A novel hybrid PSO- GWO algorithm for optimization problems. *Eng. Comput.* **35** (4), 1359–1373 (2019).
82. Muangkote, N., Sunat, K. & Chiew Chan Wattana, S. An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets. In: 2014 international computer science and engineering conference (ICSEC). IEEE. pp 209–214 (2014).
83. Long, W. et al. (20196)A novel grey Wolf optimizer algorithm with refraction learning. *IEEE Access.* **7**: 57805–57819
84. Qu, C. et al. A novel reinforcement learning based grey Wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning. *Appl. Soft Comput.* **89**, 106099 (2020).
85. Luo, K. Enhanced grey Wolf optimizer with a model for dynamically estimating the location of the prey. *Appl. Soft Comput.* **77**, 225–235 (2019).
86. Yan, F., Xu, X. & Xu, J. Grey Wolf optimizer with a novel weighted distance for global optimization. *IEEE Access.* **8**, 120173–120197 (2020).
87. Seyyed abbasi, A. & Kiani, F. 1-GWO and Ex-GWO: improved algorithms of the grey Wolf optimizer to solve global optimization problems. *Eng. Comput.* **37**, 509–532 (2019).
88. Tu, Q., Chen, X. & Liu, X. Multi-strategy ensemble grey Wolf optimizer and its application to feature selection. *Appl. Soft Comput.* **76**, 16–30 (2019).
89. Gupta, S. & Deep, K. Enhanced leadership-inspired grey Wolf optimizer for global optimization problems. *Eng. Comput.* **36**, 1777–1800 (2019b).
90. Gupta, S. & Deep, K. A novel random walk grey Wolf optimizer. *Swarm Evol. Comput.* **44**, 101–112 (2019a).
91. Sun, B. W. & Wei, S. Y. DV-hop localization algorithm based on grey Wolf optimization algorithm with adaptive adjustment strategy. *Comput. Sci.* **46** (05), 77–82 (2019).
92. Yan, F., Xu, J. Z. & Li, F. S. Training multi-layer perceptrons using chaos grey Wolf optimizer. *J. Electron. Inf. Technol.* **41** (04), 872–879 (2019).
93. Long, W. et al. (20196) A novel grey wolf optimizer algorithm with refraction learning. IEEE Access 7: 57805–57819.
94. Wang, M. N., Wang, Q. P. & Wang, X. F. Improved grey Wolf optimization algorithm based on iterative mapping and simplex method. *J. Comput. Appl.* **38** (S2), 16–20 (2018).
95. Song, P. & Liu, Y. Sky-wave radar location model based on the improved grey Wolf optimization algorithm. *Laser Opto Electron. Prog.* **56** (03), 234–242 (2019).
96. Rashid, T. A., Abbas, D. K. & Turel, Y. K. A multi-hidden recurrent neural network with a modified grey Wolf optimizer. *PLoS ONE.* **14** (3), e0213237 (2019).
97. Hicham et al. *A Novel MPPT Algorithm Based on Aquila Optimizer* (under PSC and Implementation using Raspberry, 2022).
98. Dagal, I. & Akın Transformer rail-tapped buck-boost converter design-based feedback controller for battery charging systems. Energy Storage.; e414. 2022. (2022).
99. Hicham et al. Secure and Optimized Satellite Image Sharing based on Chaotic eπ Map and Racah Moments, (2023).
100. Dagal, I. B. Akın and Akboy. A novel hybrid series Salp Particle Swarm Optimization (SSPSO) for standalone battery charging applications. Ain Shams Engineering Journal.; 13: 10174. 2022. (2022).
101. Zhijian, I. A. W. Fang et al. Hybrid SSA-PSO based intelligent direct sliding-mode control for extracting maximum photovoltaic output power and regulating the DC-bus voltage. (2023). https://doi.org/10.1016/j.ijhydene.2023.10.034
102. Zhijian, I. A. W. Fang et al. Fast DC-Link voltage regulation and maximum power extraction for standalone PV/BES system using hybrid SPSA-DSMC. (2023). https://doi.org/10.1109/ECCE53617.2023.10362571
103. Dagal, I. B. Akın and Akboy: MPPT mechanism based on novel hybrid particle swarm optimization and Salp Swarm Optimization Algorithm for battery charging through Simulink. Scientific Reports ; 12:1–17. 2022. (2022).
104. Dagal, I., Akin, B., Akboy, E. & and Improved salp swarm algorithm based on particle swarm optimization for maximum power point tracking of optimal photovoltaic systems. *Int. J. Energy Res.* **46** (7), 8742–8759 (2022).
105. Dagal, I. & Akin Burak and Yaya Dagal Dari. A modified multi-stepped constant current based on Gray Wolf algorithm for photovoltaics applications. *Electr. Eng.* https://doi.org/10.1007/s00202-023-02180-z (2024).
106. Dagal, I., Akin & Burak and Yaya Dagal Dari. An improved constant current step-based grey wolf optimization algorithm for photovoltaic systems. Journal of Intelligent & Fuzzy Systems. DOI.10.3233/JIFS-224535. (2024).
107. Nonut, A. et al. A small fixed-wing UAV system identification using metaheuristics. *Cogent Eng.* **9** (1), 2114196 (2022).
108. Tejani, G. G., Bhensdadia, V. H. & Bureerat, S. Examination of three meta-heuristic algorithms for optimal design of planar steel frames. *Adv. Comput. Des.* **1** (1), 79–86 (2016).
109. Tejani, G. G., Mashru, N., Patel, P., Sharma, S. K. & Celik, E. Application of the 2-archive multi-objective cuckoo search algorithm for structure optimization. *Sci. Rep.* **14** (1), 31553 (2024).
110. Xia, J. Y. et al. Metalearning-based alternating minimization algorithm for nonconvex optimization. *IEEE Trans. Neural Networks Learn. Syst.* **34** (9), 5366–5380 (2022).
111. Sun, G. et al. Low-latency and resource-efficient service function Chaining orchestration in network function virtualization. *IEEE Internet Things J.* **7** (7), 5760–5772 (2019).
112. Cheng, C., Deng, X., Zhao, X., Xiong, Y. & Zhang, Y. Multi-occupant dynamic thermal comfort monitoring robot system. *Build. Environ.* **234**, 110137 (2023).
113. Zhang, C. et al. A multiple model type-3 fuzzy control for offshore wind turbines using the active rotary inertia driver (ARID). *Ocean Eng.* **313**, 119337 (2024).
114. Ibrahim, A. W., Fang, Z., Li, R., Zhang, W., Xu, J., Zahir, V., … ELrashidi, A. (2024).Intelligent Nonlinear PID-Controller Combined with Optimization Algorithm for Effective Global Maximum Power Point Tracking of PV Systems. *IEEE Access*.
115. Dagal, I., Ibrahim, A. W. & Harrison, A. Leveraging a novel grey Wolf algorithm for optimization of photovoltaic-battery energy storage system under partial shading conditions. *Comput. Electr. Eng.* **122**, 109991 (2025).
116. Ibrahim, A. W., Xu, J., Al-Shamma'a, A. A., Farh, H. M. H. & Dagal, I. Intelligent adaptive PSO and linear active disturbance rejection control: A novel reinitialization strategy for partially shaded photovoltaic-powered battery charging. *Comput. Electr. Eng.* **123**, 110037 (2025).
117. Yang, M. et al. Two-stage day-ahead multi-step prediction of wind power considering time-series information interaction. *Energy* **312**, 133580 (2024).
118. Ma, S. et al. The autonomous pipeline navigation of a cockroach bio-robot with enhanced walking stimuli. *Cyborg Bionic Syst.* **4**, 0067 (2023).
119. Wang, B., Wang, Z., Song, Y., Zong, W., Zhang, L., Ji, K., … Dai, Z. (2023). A neural coordination strategy for attachment and detachment of a climbing robot inspired by gecko locomotion. *Cyborg and Bionic Systems*, *4*, 0008.
120. Yi, X., Lu, T., Li, Y., Ai, Q. & Hao, R. Collaborative planning of multi-energy systems integrating complete hydrogen energy chain. *Renew. Sustain. Energy Rev.* **210**, 115147 (2025).

121. Huang, Z., Zhou, Y., Lin, Y. & Zhao, Y. Resilience evaluation and enhancing for China's electric vehicle supply chain in the presence of attacks: A complex network analysis approach. *Comput. Ind. Eng.* **195**, 110416 (2024).
122. Zhang, C., Qiao, J., Wang, S., Chen, R., Dui, H., Zhang, Y., … Zhou, Y. (2025). Importance measures based on system performance loss for multi-state phased-mission systems. *Reliability Engineering & System Safety*, *256*, 110776.
123. Xu, X. & Wei, Z. Dynamic pickup and delivery problems with transshipments and LIFO constraints. *Comput. Ind. Eng.* **175**, 108835 (2023).
124. Zhang, Z., Bu, Y., Wu, H., Wu, L. & Cui, L. Parametric study of the effects of clump weights on the performance of a novel wind-wave hybrid system. *Renew. Energy*. **219**, 119464 (2023).
125. Peraza-Vázquez, H., Peña-Delgado, A., Merino-Treviño, M., Morales-Cepeda, A. B. & Sinha, N. A novel metaheuristic inspired by horned Lizard defense tactics. *Artif. Intell. Rev.* **57** (3), 59 (2024).
126. Abdelmalek, F., Afghoul, H., Krim, F., Bajaj, M. & Blazek, V. Experimental validation of novel hybrid grey Wolf equilibrium optimization for MPPT to improve the efficiency of solar photovoltaic system. *Results Eng.* **25**, 103831 (2025).
127. Djaballah, S. et al. A hybrid LSTM random forest model with grey Wolf optimization for enhanced detection of multiple bearing faults. *Sci. Rep.* **14** (1), 23997 (2024).
128. Deghfel, N. et al. Improving maximum power point tracking efficiency in solar photovoltaic systems using super-twisting algorithm and grey Wolf optimizer. *IET Renew. Power Gener.* **18** (15), 3329–3354 (2024).
129. Goud, B. S., Rami Reddy, C., Naga Sai kalyan, C., Udumula, R. R., Bajaj, M., Abdul Samad, B., … Kamel, S. (2022). PV/WT integrated system using the Gray Wolf Optimization Technique for power quality improvement. Frontiers in Energy Research, 10, 957971.

Author contributions

Idriss Dagal, Ibrahim AL-Wesabi: Conceptualization, Methodology, Software, Visualization, Investigation, Writing- Original draft preparation. Ambe Harrison, Wulfran Fendzi Mbasso, Ahmad O. Hourani: Data curation, Validation, Supervision, Resources, Writing - Review & Editing. Ievgen Zaitsev: Project administration, Supervision, Resources, Writing - Review & Editing.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to I.D. or I.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.