Neural Computing and Applications (2022) 34:14627–14657
https://doi.org/10.1007/s00521-022-07303-4

ORIGINAL ARTICLE

A multi-strategy random weighted gray wolf optimizer-based multi-layer perceptron model for short-term wind speed forecasting

Tufan İnaç[1] · Emrah Dokur[2] · Uğur Yüzgeç[3]

Received: 30 August 2021 / Accepted: 13 April 2022 / Published online: 10 May 2022

Abstract
Gray wolf optimizer (GWO) that is one of the meta-heuristic optimization algorithms is principally based on the hunting method and social hierarchy of the gray wolves in the nature. This paper presents the Multi-strategy Random weighted Gray Wolf Optimizer (MsRwGWO) including some effective and novel mechanisms added to the original GWO algorithm to improve the search performance. These are a transition mechanism for updating the parameter $\overrightarrow{a}$, a weighted updating mechanism, a mutation operator, a boundary checking mechanism, a greedy selection mechanism, and an updating mechanism of leader three wolves (alpha, beta, and delta wolves). We utilized some benchmark functions known as CEC 2014 test suite to evaluate the performance of MsRwGWO algorithm in this study. Firstly, during the solution of optimization problems, the MsRwGWO algorithm's behaviors such as convergence, search history, trajectory, and average distance were analyzed. Secondly, the comparison statistical results of MsRwGWO and GWO algorithms were presented for CEC 2014 benchmarks with 10, 30, and 50 dimensions. In addition, some of the popular meta-heuristic algorithms taken from the literature were compared with the proposed MsRwGWO algorithm for 30D CEC 2014 test problems. Finally, MsRwGWO algorithm was adapted to the training process of a Multi-Layer Perceptron (MLP) used in wind speed estimation and comparative results with GWO-based MLP were obtained. The statistical results of the benchmark problems and training performance of MLP model for short-term wind speed forecasting show that the proposed MsRwGWO algorithm has better performance than GWO algorithm. Source code of MsRwGWO is publicly available at https://github.com/uguryuzgec/MsRwGWO.

Keywords Gray wolf optimizer · Random weighted · Novel transition mechanism · Wind speed forecasting

1 Introduction

Meta-heuristic methods have become powerful and popular in the last thirty years due to their simple, flexible, and easy structures. Their high efficiency, easy-to-apply structure, and their ability to avoid local optimum make them widely used in today's engineering sciences. Optimization techniques aim to obtain the best solution in order to adapt the models created during scientific studies to real life. Features such as hunting techniques, feeding methods, and mating habits of various creatures in nature are frequently used in the design of meta-heuristic algorithms inspired by nature. Examples of the best known of these algorithms are Genetic Algorithm (GA) [1], Particle Swarm Optimizer (PSO) algorithm [2, 3], Artificial Bee Colony (ABC) algorithm [4, 5].

Many real-life problems may have more than one solution. Optimization techniques are also classified according to the nature of the problem to find the best solution. Meta-heuristic algorithms are classified as bio-inspired, physical, evolutionary, herd intelligence, and other nature-inspired algorithms according to their structure [6–8]. Evolutionary algorithms were inspired by Darwin's theory of natural selection [9]. Starting with a

✉ Uğur Yüzgeç
ugur.yuzgec@bilecik.edu.tr

[1] Department of Electrical and Electronics Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey

[2] Department of Electrical and Electronics Engineering, Energy Technology Application and Research Center, Bilecik Seyh Edebali University, Bilecik, Turkey

[3] Department of Computer Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey

random population, these algorithms update the population through evolutionary mechanisms such as mutation and crossover. Differential Evolution (DE) algorithm [10, 11] and Genetic Algorithm (GA) [1, 12] are the best known examples of evolutionary algorithms. Physical algorithms are created and developed inspired by physical phenomena in nature. These algorithms, which start with a single solution, are replaced by a number of physical equations during iterations. Shuffled Frog Leaping algorithm (SFLA) [13, 14], Harmony Search (HS) algorithm [15, 16], Tabu Search (TS) algorithm [17], and Simulated Annealing (SA) algorithm [18] are one of the most popular examples. Swarm intelligence approaches are algorithms that mimic collective intelligence such as birds' flocks, bee colonies, and fish flocks, which are composed of dispersed individuals but collectively, act together by interacting with each other [6]. The most well-known examples of these algorithms are; Kennedy and Eberhart's Particle Swarm Optimizer (PSO) algorithm [2, 3], Artificial Bee Colony (ABC) algorithm of Karaboga et.al [4, 19], Ant Colony Optimizer (ACO) [20, 21], and Fish Swarm Algorithm (FSA) of Li et al. [22]. Bio-inspired algorithms contain natural meta-heuristics derived from the movements of living organisms. The most popular examples of such algorithms are the Artificial Immune (AI) algorithm [23, 24] and the Bacterial Foraging Optimization (BFO) algorithm [25, 26]. There are many studies on other meta-heuristic algorithms inspired by nature, applications for solving real optimization problems. Some of these studies are as follows: Gravitational Search Algorithm (GSA) [27, 28], Biogeography-Based Optimizer (BBO) [29, 30], Invasive Weed Optimization (IWO) algorithm [31], Sine–Cosine optimization Algorithm (SCA) [32], Cuckoo Search (CS) algorithm [33, 34], Harris Hawks Optimization (HHO) algorithm [35, 36], Cultural Algorithm (CA) [37, 38], Antlion Optimization (ALO) algorithm [39, 40], Fruit Fly Optimization Algorithm (FFOA) [41], Gray Wolf Optimization (GWO) algorithm [42, 43], Grasshopper Optimization Algorithm (GOA) [44], Imperialist Competitive Algorithm (ICA) [45, 46], Firefly Algorithm (FA) [47, 48], Moth-Flame Optimization (MFO) algorithm [49], Dragonfly Algorithm (DA) [50, 51], and Whale Optimization Algorithm (WOA) [52].

One of the recently popular meta-heuristic algorithms is the Gray Wolf Optimization (GWO) algorithm [42]. It is a meta-heuristic approach that imitates the hunting patterns and leadership hierarchy of gray wolves belonging to the gray wolves (canis lupus) family in nature. Gray wolves are predatory species at the top of the food chain in natural life. They move in packs of 5–12 Gy wolves. Wolves are managed with a very rigid and dominant hierarchical structure. It has a structure in which gray wolves in the group are called alpha, beta, omega, delta, and dominated from top to bottom. The dominant member of the leading gray wolf pack is the alpha wolf. The alpha wolf is not always the strongest member of the wolf group; it is the best in terms of its ability to lead the group. Alpha wolf usually includes hunting, sleeping place, waking time, etc., in the wolf pack. It is responsible for making decisions on matters. The beta wolf, which ranks second in the ranking, helps alpha in decision making and other activities. While the beta wolf is hierarchically linked to the alpha wolf, it also rules the others. The lowest category omega wolf is submissive to all of the other dominant wolves. Gray wolves in the group are called delta if not alpha, beta, and omega wolves. While delta wolves are hierarchically linked to alpha and beta wolves, they dominate omega wolves [42]. This hierarchy is presented in detail in the next chapters. The GWO algorithm currently applied to many real-world problems, such as power system load forecasting [53], robot path planning [54], feature selection in classification problems [55], optimal control of DC motor [56], hyperspectral band selection [57], multilevel image thresholding [58], short-term photovoltaic output forecasting in solar energy [59], and wind speed forecasting [60] etc.

As can be seen, one of the application areas of the GWO algorithm is wind speed forecasting studies. The importance of wind energy systems among renewable energy sources is increasing rapidly today. The short-term estimation of the electrical energy to be obtained from wind energy conversion systems is of great importance in terms of planning, reliability, and management of power systems. One of the most important parameters of the energy to be obtained from wind energy systems is wind speed. Due to the discrete, chaotic, and non-stationary nature of wind speed data, meta-heuristic algorithms proposed in this field in the literature in wind speed estimation and their hybrid approaches are developed. Niu et al. [61] have improved performance of the wind speed forecasting by using optimal feature selection and an artificial neural network optimized by a modified bat algorithm. Liu et al. [60] proposed a hybrid model for multi-step wind speed estimation by optimizing Regular Extreme Learning Machine (RELM) parameters with the Gray Wolf Optimization (GWO) algorithm. Xiao et al. [62] proposed a unified model based on the Chaotic Particle Swarm Optimizer (CPSO) algorithm to optimize the weight coefficients in wind speed estimation. Zhang et al. [63] verified wind series from four separate wind fences using a modified Flower Pollination Algorithm (FPA). Wang et al. [64] proposed a new hybrid system for wind speed estimation using the Multi-Objective Whale Optimization Algorithm (MOWOA). Osorio et al. [65] Evolutionary Particle Swarm Optimizer (EPSO) algorithm-based Fuzzy Inference System (ANFIS) has achieved less uncertainty as well as low computation load in short-term wind speed estimation with its hybrid approach. Fei and He [66] proposed a hybrid model of wavelet decomposition and a new hybrid model based on Artificial Bee Colony (ABC) algorithm. Rahmani et al. [67] proposed a new hybrid model for short-term wind energy prediction from the hybridization of the Ant Colony Optimizer (ACO) and the Particle Swarm Optimizer (PSO) algorithm. Altan et al. [68] developed a reliable and accurate new method of wind speed estimation based on Long Short Term Memory (LSTM) network and Gray Wolf Optimizer (GWO) algorithm and decomposition methods. Fu et al. [69] proposed a mutation and hierarchy-based hybridization strategy of hybrid Harris Hawk Optimizer (HHO) and Gray Wolf Optimizer (GWO) for multi-step forward short-term wind speed estimation. Wang et al. [70] developed a hybrid Elman Neural Network (ENN) method optimized with Multi-objective Gray Wolf Optimization (MOGWO) algorithm for short-term wind speed prediction. Wu et al. [71] proposed a hybrid system with multipurpose optimization using the Extreme Learning Machine (ELM) optimized by the Multi-objective Gray Wolf Optimization (MOGWO) algorithm in wind speed prediction. Barman and Choudhury proposed a Support Vector Machine (SVM) hybrid power system load forecasting method hybridized with the similarity-based Gray Wolf Optimization (GWO) algorithm for use in abnormal power system situations in Assam, India [53]. Singh and Dhillon [72] developed the hybrid algorithm called the Ameliorated Gray Wolf Optimization (AGWO) algorithm to solve the economic load distribution problem and validated it in benchmarking problems for medium-sized electric generator systems. Pradhan et al. [73] applied it to nonlinear economic load distribution problems such as valve point effect, ramp speed, and restricted zone to justify the effectiveness of the Gray Wolf Optimization (GWO) algorithm. Jayabarathi et al. [74] proposed the Hybrid Gray Wolf Optimization (HGWO) algorithm, which they developed in the solution of economic distribution problems of power systems. Pradhan et al. [75] used their proposed hybrid Oppositional Gray Wolf Optimization (OGWO) algorithm in the solution of economic load distribution problems and made comparative studies with the Gray Wolf Optimization (GWO) algorithm to examine its effectiveness.

The motivation of our research is to improve the search performance of one of the popular heuristics, the gray wolf optimization algorithm, and adapt it to a real-world problem. The main purpose of this study is to first improve the performance of the gray wolf optimization algorithm and then use it to tune the artificial neural network model parameters for short-term wind speed forecasting, which is a real-world optimization problem. In this study, an improved version of GWO which is called Multi-strategy Random weighted Gray Wolf Optimizer (MsRwGWO) is presented which has six different mechanisms to improve search ability of the original GWO algorithm. These are a transition mechanism for updating $\vec{a}$ parameter, a novel random weighted updating mechanism, a mutation operator, a new boundary checking mechanism, a greedy selection mechanism, and renewed update mechanism of alpha, beta, and delta wolves. In this paper, the proposed MsRwGWO is analyzed in terms of convergence, search history, trajectory, and average distance. The performance of the MsRwGWO is examined in detail with benchmark functions known as CEC 2014. In addition, the MsRwGWO-based Multi-Layer Perceptron (MLP) approach is compared with GWO-MLP hybrid model for real-world problem like wind speed forecasting.

Section 2 presents traditional GWO architecture. The features of the proposed meta-heuristic approach, MsRwGWO, are presented in Sect. 3. The analysis of MsRwGWO is given in Sect. 4. In addition, proposed MsRwGWO-based MLP results are presented for wind speed forecasting comparatively in the same chapter. Finally, conclusions are given in Sect. 5.

2 Gray wolf optimizer (GWO)

The gray wolf optimization algorithm (GWO) is an optimization algorithm that mimics the hunting strategy and the social leadership of gray wolves proposed by Mirjalili [42]. Gray wolves mostly prefer to live as a group. The average group size is between 5 and 12 wolves. The hierarchy of gray wolves is in the form of four groups: alpha, beta, delta, and omega wolves. The leader or dominant wolf is called the alpha wolf, and the alpha wolf is the best wolf to manage other wolves in the group and is usually responsible for deciding on waking time, sleeping place, hunting, and so on. The second in the social hierarchy of the wolf group is the beta wolf. The beta wolf is the assistant of the alpha wolf in many events. The delta wolf is the third obliged to obey alpha and beta wolves and can only dominate omega wolves. The omega is the lowest level gray wolf [42]. The gray wolf hierarchy is shown in Fig. 1.

Another social behavior of gray wolves is group hunting strategy. In this strategy, gray wolves firstly recognize the location of the prey and surround the prey under the leadership of the alpha wolf. In the mathematical model of the hunting strategy of gray wolves, it is assumed that alpha, beta, and delta wolves have better information about the location of the prey. Therefore, the first three best solutions (alpha, beta, and delta wolves) are used to update the positions of the wolves in the GWO algorithm. The rest

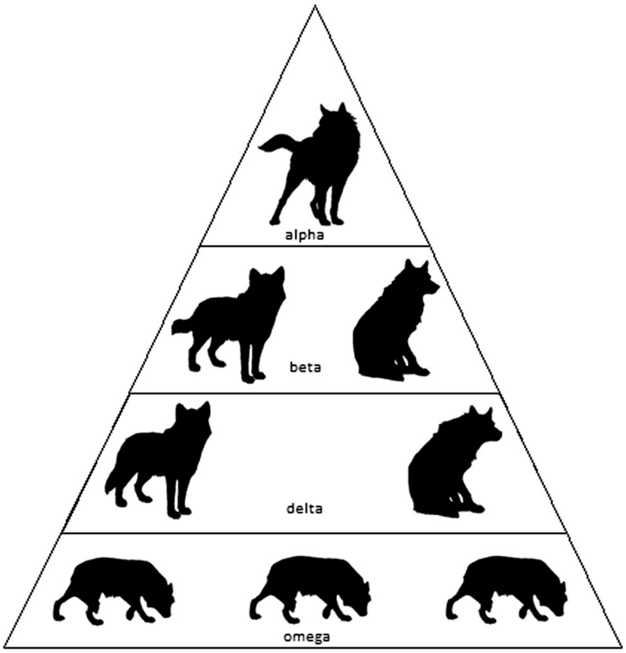


Fig. 1 The social hierarchy of gray wolves [54]

of the wolves are assumed to be omega wolves [42]. The omega wolves follow the alpha, beta, and delta wolves during the hunt. The hunting mechanism of gray wolves is modeled using equations as given below:

$$\vec{D}_\alpha = \left|\vec{C}_\alpha \cdot \vec{X}_\alpha - \vec{X}_i\right| \tag{1}$$

$$\vec{D}_\beta = \left|\vec{C}_\beta \cdot \vec{X}_\beta - \vec{X}_i\right| \tag{2}$$

$$\vec{D}_\delta = \left|\vec{C}_\delta \cdot \vec{X}_\delta - \vec{X}_i\right| \tag{3}$$

$$\vec{U}_\alpha = \vec{X}_\alpha - \vec{A}_\alpha \vec{D}_\alpha \tag{4}$$

$$\vec{U}_\beta = \vec{X}_\beta - \vec{A}_\beta \vec{D}_\beta \tag{5}$$

$$\vec{U}_\delta = \vec{X}_\delta - \vec{A}_\delta \vec{D}_\delta \tag{6}$$

$$\vec{X}_i = \left(\vec{U}_\alpha + \vec{U}_\beta + \vec{U}_\delta\right)/3 \tag{7}$$

where $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$ denote the distance vector between gray wolf (alpha, beta, and delta) and prey, $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ represent the position vector of the prey for alpha, beta, delta wolves, $\vec{X}_i$ indicates the gray wolf (omega) position vector at ith iteration, $\vec{U}_\alpha, \vec{U}_\beta, \vec{U}_\delta$ stand for the trial vector for alpha, beta, delta gray wolves, $\vec{C}_\alpha, \vec{C}_\beta, \vec{C}_\delta, \vec{A}_\alpha, \vec{A}_\beta, \vec{A}_\delta$ are the coefficient vectors for alpha, beta, delta wolves. These vectors are found according to the equations given below:

$$\vec{A}_i = 2\vec{a}\vec{r}_{i1} - \vec{a}, i = \alpha, \beta, \delta \tag{8}$$

$$\vec{C}_i = 2\vec{r}_{i2}, i = \alpha, \beta, \delta \tag{9}$$

where $\vec{a}$ stands for a vector linearly decreased from 2 to 0, and $\vec{r}_{i1}$ and $\vec{r}_{i2}$ indicate the random vector in [0,1]. Figure 2 shows the hunting strategy of gray wolves. As can be seen in this figure, each gray wolf in the group updates its position according to the distance between the alpha, beta, and delta gray wolves and gets closer to the prey. Eventually, the prey is caught by gray wolves and the wolf group finishes the hunt by attacking the prey [42]. The pseudocode of the original GWO algorithm is given in Algorithm 1.

Algorithm 1: Pseudo code of GWO algorithm.

```
Initialize the positions of gray wolves
Calculate the fitness values of gray wolves
Save the best gray wolf as alpha
Save the second best gray wolf as beta
Save the third best gray wolf as delta
WHILE (iteration < maximum iteration)
    FOR each gray wolf
        Generate the coefficient vectors for alpha, beta, and delta
        Calculate the distance vectors ($\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$)
        Calculate the trial vectors ($\vec{U}_\alpha, \vec{U}_\beta, \vec{U}_\delta$)
        Update the position of omega gray wolf using Eq. (7)
    END FOR
    Calculate the fitness values for updated all gray wolves
    FOR each gray wolf
        IF (gray wolf < alpha)
            update alpha
        ELSE IF (gray wolf < beta)
            update beta
        ELSE IF (gray wolf < delta)
            update delta wolf
        END IF
    END FOR
         Increase iteration
 END WHILE
 return alpha
```

3 Multi-strategy random weighted gray wolf optimizer (MsRwGWO)

In this study, we propose some novel approaches to develop the search performance of the original GWO algorithm. These proposed new approaches are mentioned in this section. Six different mechanisms were added to the original GWO algorithm, and they are as follows, respectively:

1. A transition mechanism was adapted for updating the parameter $\overrightarrow{a}$ used in Eq. (8),

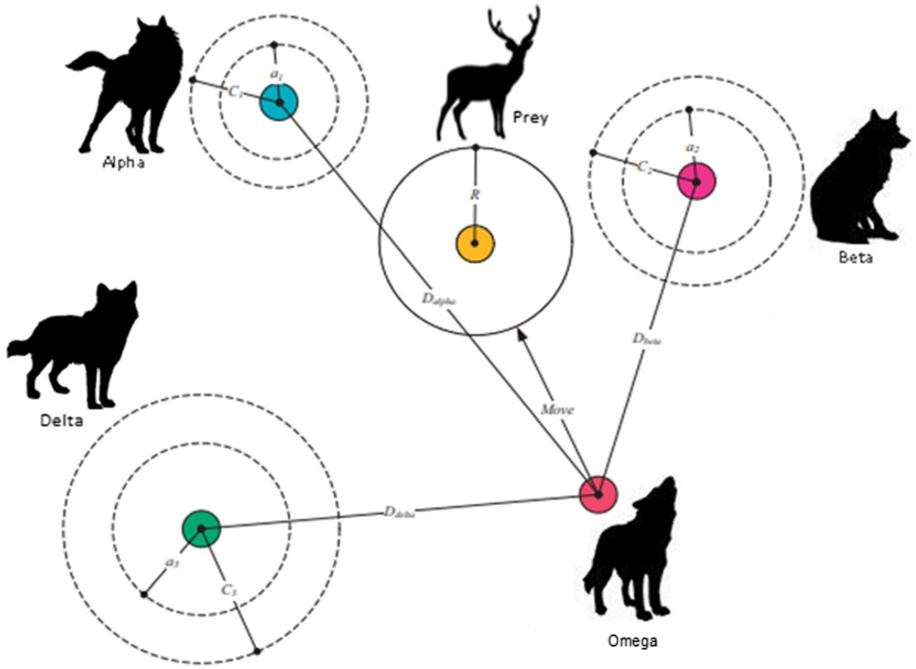


Fig. 2 The hunting mechanism of gray wolves [54]

2. A new weighted updating mechanism was presented for updating the positions of wolves,
3. A mutation operator was added into the GWO algorithm,
4. A novel mechanism was used for checking boundaries of the search space,
5. A selection mechanism has been added to the algorithm,
6. The update mechanism of alpha, beta, and delta wolves was renewed.

We named the proposed algorithm the Multi-strategy Random weighted Gray Wolf Optimizer (MsRwGWO) because of these six different mechanisms added to the original GWO algorithm. The original GWO algorithm has some parameters, and one of them is the parameter $\overrightarrow{a}$ linearly decreased from 2 to 0 during the optimization process. This parameter plays an important role in the transition from the exploration phase to the exploitation phase. The higher values of this parameter enable the global exploration, while the low values of its enable the local exploitation of the search space. Although this parameter decreases linearly in the original GWO algorithm, nonlinear changes in the exploration and exploitation behaviors of an algorithm are needed to keep away from local optimal solutions in many problems. A suitable selection of this parameter is very important for the balance between exploration and exploitation phases. In this study, this transition is redefined according to a nonlinear function proposed by Gupta and his friends for Sine Cosine optimizer in 2020 [76] to avoid local optimal solutions. The proposed nonlinear transition function for the parameter $\vec{a}$ is given below:

$$\vec{a} = 2 \times \sin\left(\left(1 - \frac{\text{iter}}{\text{Max\_iter}}\right) \times \frac{\pi}{2}\right) \tag{10}$$

Figure 3 shows the changes of the original linear transition parameter in GWO algorithm and the proposed nonlinear transition parameter together. The higher values of the parameter $\vec{a}$ facilitate the exploration phase ($\vec{a} > 1$), while the lower values of the parameter $\vec{a}$ facilitate to the local exploitation phase ($\vec{a} < 1$) of the search space. From Fig. 3, the transition procedure allows that the duration of the exploration (about 65%) is a little bit longer than the duration of the exploitation (about 35%). Thus, it is

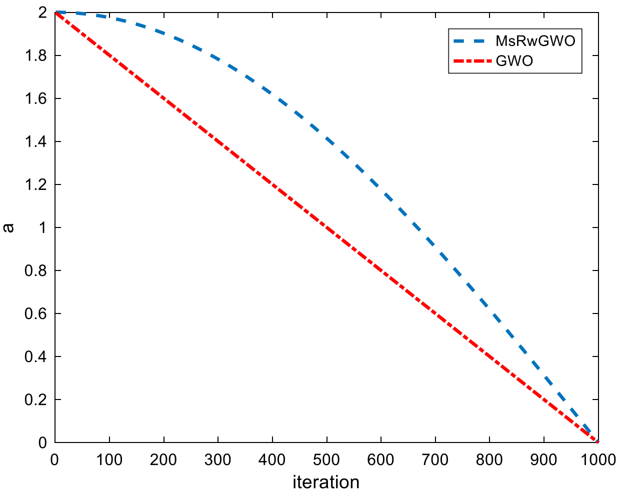


Fig. 3 Original and the proposed transition parameters

predicted that the transition between both phases is better during optimization process.

After the use of nonlinear transition function on change of the parameter $\vec{a}$ in GWO algorithm, we focused on the mechanism of updating the positions of the gray wolves. In the original GWO algorithm, the positions of gray wolves are updated by averaging the trial vectors $(\vec{U}_\alpha, \vec{U}_\beta, \vec{U}_\delta)$ calculated according to the positions of the alpha, beta, and delta gray wolves. In the proposed updated mechanism, the new positions of the wolves in the group are determined according to the fitness scores of the alpha, beta, and delta wolves. The equations of this mechanism are given as follows:

$$S = \sum_{i=\alpha,\beta,\delta} \frac{1}{f\left(\vec{X}_i\right)} \tag{11}$$

$$w_\alpha = \frac{f\left(\vec{X}_\alpha\right)^{-1}}{S}, w_\beta = \frac{f\left(\vec{X}_\beta\right)^{-1}}{S}, w_\delta = \frac{f\left(\vec{X}_\delta\right)^{-1}}{S} \tag{12}$$

$$\vec{X}_i = w_\alpha \vec{U}_\alpha + w_\beta \vec{U}_\beta + w_\delta \vec{U}_\delta \tag{13}$$

where S denotes the sum scores of the alpha, beta, and delta wolves, $f\left(\vec{X}_i\right)$ represents the fitness value of the $\vec{X}_i$ solution (i denotes the indices of the alpha, beta, and delta wolves), and this means the objective function's value in a mathematical optimization problem. $w_\alpha, w_\beta, w_\delta$ indicate the score weight values of the alpha, beta, and delta wolves. These score weights are utilized for updating the position of gray wolves ($\vec{X}_i$). Thus, instead of averaging the trial vectors in updating each gray wolf's position, each position is updated by the weighted sum of the trial vectors according to the score weights of three leader gray wolves.

In this new update mechanism, firstly, the total score value (S) is calculated for the three leader wolves (alpha, beta, and delta) based on their fitness values. Then, the score weight value of each wolf is found according to the fitness values of these three leader wolves. Thus, the score weights of the three leader wolves are used to update the positions of gray wolves in proportion to their fitness values. Here, the alpha gray wolf's score weight is higher than the other two gray wolfs' score weights, just as the beta gray wolf is higher than the delta, so the alpha leader wolf contributes more to updating the wolf's positions than the beta and delta wolves. Figure 4 shows the new update mechanism of gray wolves. This new updated mechanism helps to improve exploration and exploitation abilities of the algorithm. Although these proposed innovations work well on some problems, the algorithm may still get stuck at the local optimal point in some cases. Therefore, a mutation operator was added to this algorithm for situations where better positions for wolves cannot be found by the proposed update mechanism and nonlinear transition parameter. The mutation mechanism is given below:

$$\vec{X}_i(t+1) = \vec{X}_i(t) + 0.1 \times \left(\vec{U}_b - \vec{L}_b\right) \times r_m \tag{14}$$

where $\vec{U}_b$ represents the upper boundary, $\vec{L}_b$ is the lower boundary of the position of the search agent, and r_m stands for the normally distributed random number.

In the original GWO algorithm, it is checked whether the positions of the wolves exceed the search space boundaries after updating. If the new position of the gray wolf exceeds the upper or lower boundaries, the gray wolf's position is equalized to the boundary value to prevent exceeding the boundary conditions. This situation is quite common during the exploration phase, and therefore, a lot of wolves get stuck on the boundaries of the search space in some problems. To avoid this situation, we proposed a novel boundary checking mechanism. In this control procedure, if the boundary constraint is violated, the new position of the gray wolf is set to be the middle of its previous position and boundary value of the search space. The proposed boundary checking procedure is given as follows:

$$if(\vec{X}_i(t) > \vec{U}_b) \Rightarrow \vec{X}_i(t) = \frac{\vec{X}_i(t-1) + \vec{U}_b}{2} \tag{15}$$

$$if(\vec{X}_i(t) < \vec{L}_b) \Rightarrow \vec{X}_i(t) = \frac{\vec{X}_i(t-1) + \vec{L}_b}{2} \tag{16}$$

where $\vec{L}_b$ and $\vec{U}_b$ stand for the lower and upper boundary values, and t represents the current iteration. The original GWO algorithm has no selection mechanism. A simple selection mechanism was implemented to the GWO algorithm for the use of gray wolves that are more suitable in the population in later iterations. This selection mechanism is given below:

$$\begin{aligned} if(f(\vec{X}_i(t-1)) < f\left(\vec{X}_i(t)\right)) \Rightarrow \vec{X}_i(t) \\ = \vec{X}_i(t-1) \wedge f\left(\vec{X}_i(t)\right) \\ = f\left(\vec{X}_i(t-1)\right) \end{aligned} \tag{17}$$

where $\vec{X}_i(t-1)$ denotes the old position of ith gray wolf, and $\vec{X}_i(t)$ represents the updated position of ith gray wolf. The latest development on GWO in this study is on the update mechanism of alpha, beta, and delta wolves. In the original GWO algorithm, the fitness value of each gray wolf whose position is updated is compared with the fitness values of alpha, beta, and delta wolves one by one, and updating positions of alpha, beta, and delta wolves is performed. Here we observed that after updating the positions of alpha, beta, or delta wolves, the old position values of these wolves were not used. The new updating mechanism of the positions of alpha, beta, and delta wolves is shown in Fig. 5.

In Fig. 5, omega wolf represents the gray wolf whose its position is updated. Unlike the original GWO algorithm, in updating the alpha wolf, if the fitness value of the omega wolf is better than that of the alpha wolf, the new alpha wolf becomes the omega wolf, also the old alpha wolf is updated as the new beta wolf, and the old beta wolf is updated as the new delta wolf. Likewise, in the updating the beta wolf, the old beta wolf becomes the new delta wolf. The pseudocode of the proposed MsRwGWO algorithm is given in Algorithm 2.

4 Results and discussion

First, different metrics were examined for the analysis of the proposed MsRwGWO algorithm. These are as follows: convergence, search history, trajectory, and average distance. The main purpose of these analyses is to reveal the search behavior of the proposed MsRwGWO algorithm during the optimization process and compare it with that of the original GWO. The used metrics are the position of gray wolves from the first to the last iteration (search history), the position of the best gray wolf (alpha wolf) in each iteration (trajectory), the mean distance of the first gray wolf's position to the others in the group (average

Algorithm 2: Pseudo code of MsRwGWO algorithm.

```
Initialize the positions of gray wolves
Calculate the fitness values of gray wolves
Save the best gray wolf as alpha
Save the second best gray wolf as beta
Save the third best gray wolf as delta
WHILE (iteration < maximum iteration)
    The parameter 'a' is calculated by using modified Eq. (10)
    FOR each gray wolf
        Generate the coefficient vectors for alpha, beta, and delta
        Calculate the distance vectors ($\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$)
        Calculate the trial vectors ($\vec{U}_\alpha, \vec{U}_\beta, \vec{U}_\delta$)
        Calculate the weight values of alpha, beta, and delta using Eqs. (11-12)
        Update the position of gray wolf using weighted total in Eq. (13)
        Apply the mutation operator to the gray wolf in Eq. (14)
        Boundary check for the updated gray wolf using Eqs. (15-16)
    END FOR
    Calculate the fitness values for updated gray wolves
    FOR each gray wolf
        IF (gray wolf < alpha)
            delta = beta
            beta = alpha
            update alpha
        ELSE IF (gray wolf < beta)
            delta = beta
            update beta
        ELSE IF (gray wolf < delta)
            update delta
        END IF
    END FOR
    Increase iteration
END WHILE
return alpha
```

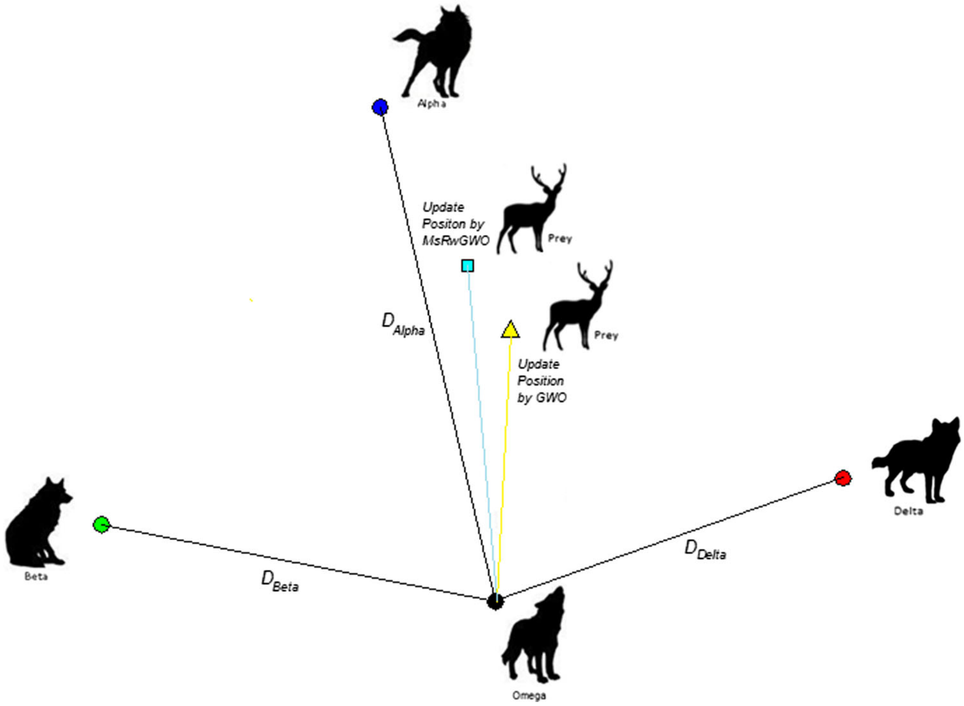


Fig. 4 Update mechanism of MsRwGWO algorithm

distance), and the fitness of the best gray wolf (alpha wolf) obtained from the first to the last iteration (convergence).

We used some benchmark functions known as CEC2014 from the literature to make these analyses. After the analysis of the MsRwGWO algorithm, thirty benchmark problems based on the IEEE Congress on Evolutionary Computation (CEC) 2014 test suite were addressed [77]. For the different problem dimensions, these benchmark problems were solved by the original GWO algorithm and the MsRwGWO algorithm and the results were compared statistically. Furthermore, the superiority of the developed structure was shown by comparing the 30D CEC 2014 test problem results of the proposed MsRwGWO algorithm with up-to-date meta-heuristic algorithms. In the last subsection of this section, we tested GWO and MsRwGWO algorithms for short-term forecasting of wind speed by integrating them into a Multi-Layer Perceptron (MLP) structure.

4.1 Analyses of the proposed MsRwGWO algorithm

In this study, the MsRwGWO algorithm was realized to improve exploration and exploitation capabilities of the original GWO algorithm. To show convergence behavior of the MsRwGWO algorithm, we considered analyses for four different metrics. First is convergence analysis, second is search history analysis, third is trajectory analysis, and the last is average distance analysis. Four different benchmark functions were selected among the CEC 2014 benchmark problems to perform these analyses. CEC 2014 test suite includes four types of problems: simple multimodal, unimodal, composition, and hybrid functions. Table 1 summarizes these benchmark functions and their names. The detailed information about CEC 2014 benchmark problems can be found from the paper of Liang et al. [77]. The functions used for analysis are as follows: rotated bent cigar (FN2) function from unimodal functions, shifted and rotated Weierstrass function (FN6) and shifted and rotated expanded Scaffer function (FN16) from simple multimodal functions, and composition function 1 ($N = 5$) from composition functions. Figure 6 shows the benchmark functions used in analyses of the proposed MsRwGWO algorithm. Each subfigure includes 3D maps and contour lines for 2D of these functions. As an important note, the initial positions of the gray wolves were taken equally for all analyses in both algorithms.

4.1.1 Convergence analysis

The performance of a meta-heuristic algorithm depends on its convergence behavior that occurs in the solution of an optimization problem. Convergence behavior gives us information about the speed of the algorithm. In this context, the convergence curves of the proposed MsRwGWO and original GWO algorithms were obtained for solving four different benchmark problems. In this analysis, the problem dimension was taken as 2. Figure 7 shows the comparison results of the convergence behaviors of both algorithms. These curves show the error values of the best

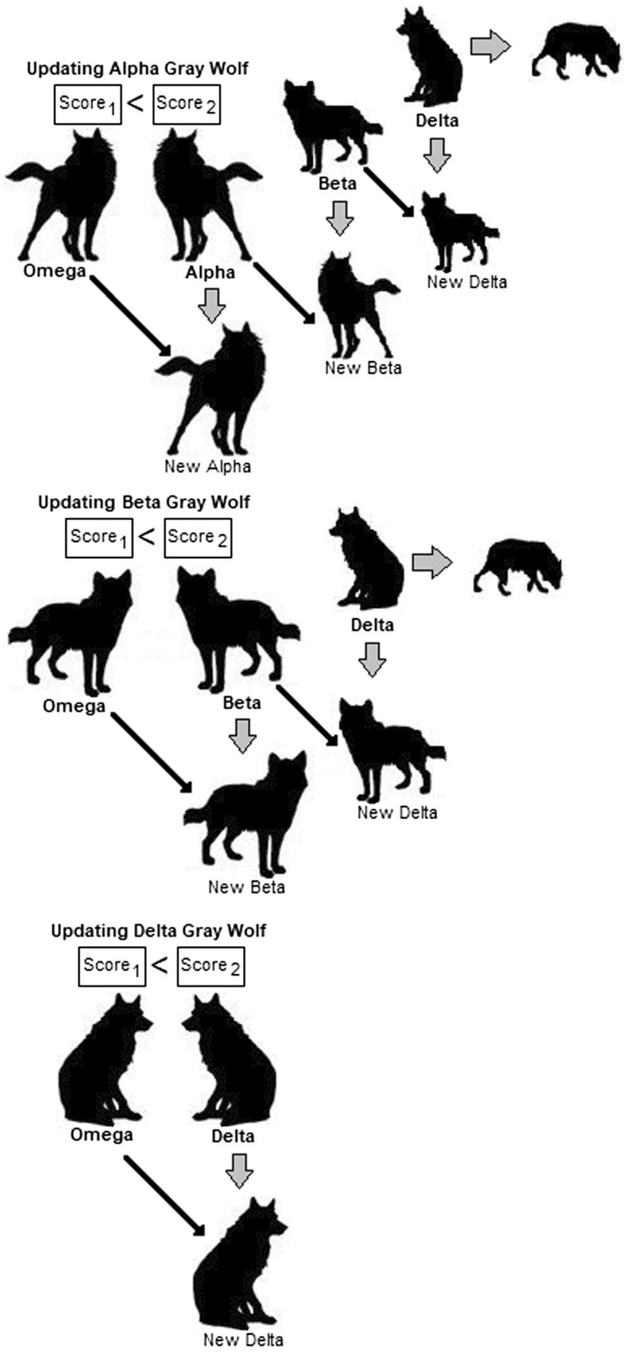


Fig. 5 The proposed new updating mechanism of alpha, beta, and delta wolves

gray wolves found by the MsRwGWO and GWO algorithms throughout the optimization. Here, these error values are calculated by taking the difference between the best-found solution and the real solution of the problem.

In Fig. 7, the convergence curves of the original GWO and MsRwGWO algorithms were given on the logarithmic scale. Looking at the trends in the convergence curves, it is seen that MsRwGWO has a faster convergence than the GWO algorithm in solving four different benchmark problems. The convergence curves show that the gray wolves in the population cooperate to improve the search performance by updating their current positions better, thanks to the proposed new and effective mechanisms, while searching for the global solution point in the optimization problem. Looking at the convergence results for each test function, it can be said that the MsRwGWO algorithm shows approximately the same trend as the GWO until the exploitation phase in the FN2 and FN16 benchmarks, which have more flat surfaces, but MsRwGWO algorithm shows better convergence behavior in the exploitation phase. The convergence curves obtained for the FN6 function, which consists of abundant peaks and troughs, show us that the convergence behavior of the GWO algorithm in the exploration phase is better, and the MsRwGWO algorithm is better in the exploitation phase. The FN23 function convergence curve shows that the MsRwGWO algorithm has a better convergence capability than GWO in both phases. The analysis results show the ability of the proposed MsRwGWO to find a solution closer to the global optimum.

Rapid descents are observed in the convergence curves obtained by the proposed MsRwGWO algorithm in the exploitation and exploration phases. This is due to the new weighted update mechanism presented for the MsRwGWO algorithm. The strength of this new mechanism is that the positions of gray wolves are updated according to the score weights of the three leader wolves in proportion to their fitness values.

4.1.2 Search history analysis

In this analysis, we examined the search history, which gives the movements of search agents (gray wolves) in the search space during the solution of the optimization problem. In Fig. 8, the search history results of the gray wolves obtained by GWO and MsRwGWO algorithms are shown for some benchmark functions selected from CEC 2014 test suite. Here, these analyses were performed by taking the initial gray wolf positions (initial population) the same for all functions. The positions of the updated gray wolves are shown on the contour surfaces of the benchmark functions at every 100 steps of the iteration. We used the number of gray wolves as 20 in this analysis. The results of this analysis show that the distribution of gray wolves around the global optima, which are updated by the MsRwGWO algorithm, is higher than the distribution of gray wolves updated by the GWO algorithm in the search space during the exploration and exploitation phases.

So, it is possible to say that the MsRwGWO searches the most promising areas of the search space in the phases of exploration and exploitation. It is seen that the positions of the gray wolves found by the GWO algorithm are stuck on

Table 1 CEC 2014 benchmark functions

Nos.	Function name	Solution	No	Function name	Solution
FN1	High cond. elliptic with rotated	100	FN16	Shifted and rotated expanded Scaffer's F6	1600
FN2	Bent cigar with rotated	200	FN17	Hybrid Func. 1 ($N \rightarrow 3$)	1700
FN3	Discus with rotated	300	FN18	Hybrid Func. 2 ($N \rightarrow 3$)	1800
FN4	Rosenbrock with shifted and rotated	400	FN19	Hybrid Func. 3 ($N \rightarrow 4$)	1900
FN5	Ackley with shifted and rotated	500	FN20	Hybrid Func. 4 ($N \rightarrow 4$)	2000
FN6	Weierstrass with shifted and rotated	600	FN21	Hybrid Func. 5 ($N \rightarrow 5$)	2100
FN7	Griewank with shifted and rotated	700	FN22	Hybrid Func. 6 ($N \rightarrow 5$)	2200
FN8	Rastrigin with shifted	800	FN23	Composition Func. 1 ($N \rightarrow 5$)	2300
FN9	Rastrigin with shifted and rotated	900	FN24	Composition Func. 2 ($N \rightarrow 3$)	2400
FN10	Schwefel with shifted	1000	FN25	Composition Func. 3 ($N \rightarrow 3$)	2500
FN11	Schwefel with shifted and rotated	1100	FN26	Composition Func. 4 ($N \rightarrow 5$)	2600
FN12	Katsuura with shifted and rotated	1200	FN27	Composition Func. 5 ($N \rightarrow 5$)	2700
FN13	HappyCat with shifted and rotated	1300	FN28	Composition Func. 6 ($N \rightarrow 5$)	2800
FN14	GHBat with shifted and rotated	1400	FN29	Composition Func. 7 ($N \rightarrow 3$)	2900
FN15	F4 with expanded plus F7	1500	FN30	Composition Func. 8 ($N \rightarrow 3$)	3000

the boundary values of the search space, especially on the surfaces of the benchmark problems except for the FN6 test problem. This is because gray wolves that exceed the limit values during the exploration phase are positioned on the boundary values. Thanks to the proposed new boundary control mechanism, this does not appear in the MsRwGWO's results.

4.1.3 Trajectory analysis

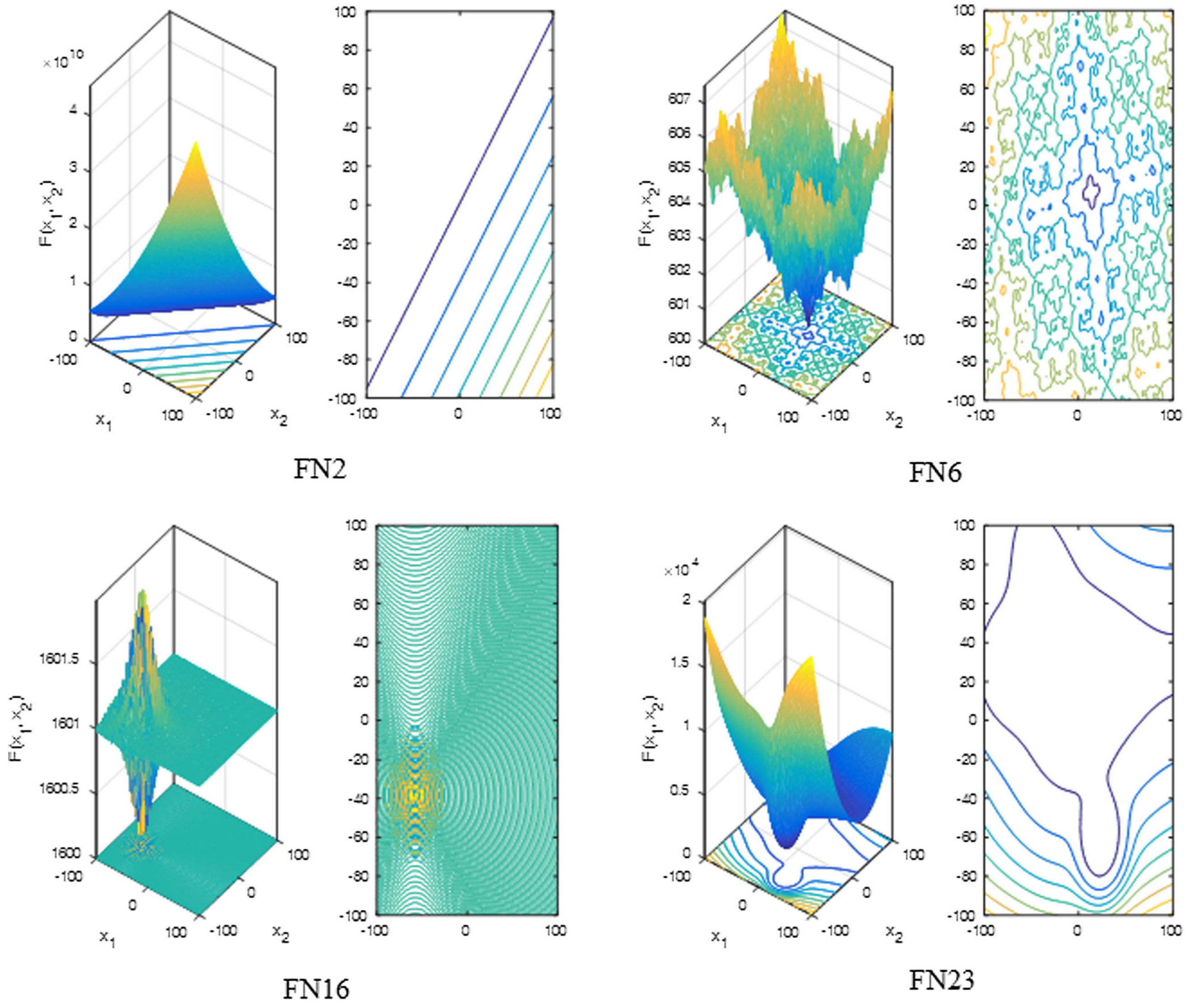
In this analysis, we examined how the position of the best gray wolf (alpha wolf) in each iteration changes in the search space of the problem during the solution of the optimization problem. The results of the trajectory analysis for the selected benchmark functions are shown in Figs. 9, 10, 11 and 12. There are two graphs in each figure: First gives changes in the position of the alpha gray wolf (elite candidate) on the contour surface of the search area during optimization process, and second shows the position of the alpha gray wolf separately for two dimensions.

In the graphic to the right of the figures containing the analysis results, the red markers indicate the positions of best gray wolves (alpha) obtained by both algorithms at the end of the optimization. Looking at the trajectory analysis results of FN2, FN6, and FN16 benchmark problems, the positions of the alpha wolves obtained by GWO and MsRwGWO algorithms at the end of the optimization process are found to be very close to the global optimum and approximately the same. In the analysis of the FN2 benchmark problem, although the position changes of the alpha wolves show different trends for both algorithms, as a result, the positions of the alpha wolves are obtained to be very close to each other at the end of the optimization.

The only different result of the trajectory analysis between GWO and MsRwGWO algorithms is seen in the FN23 benchmark problem. Here, we come across two different elite solutions at the end of the optimization process. The original GWO algorithm cannot find an alpha wolf (best candidate) close to the global optima, namely, it gets stuck in the local minima for this benchmark. The proposed MsRwGWO algorithm presents an alpha wolf as elite search agent closer to the global optima of the FN23 benchmark. Shortly, from the analysis results of MsRwGWO algorithm, the alpha wolf's position is faster updated in the exploration stage and it gets closer to global optima in the exploitation stage.

4.1.4 Average distance analysis

Average distance analysis gives the mean distance of the first gray wolf's position to the others in the group during optimization process. This shows the exploratory or exploitative behaviors of the MsRwGWO algorithm. Figure 13 shows the average distance analysis results of the proposed MsRwGWO and the original GWO algorithms for the selected benchmarks. As can be seen from the analysis result of the MsRwGWO, the average distance trends of the gray wolves have less oscillation and fluctuation compared to that in the GWO algorithm thanks to the selection mechanism added into the GWO algorithm and the new update mechanism of the alpha, beta, and delta wolves. Looking at the analysis results of the FN23 and


FN2

FN6

FN16

FN23

Fig. 6 Benchmark functions used for analyses of the proposed MsRwGWO algorithm

FN2 benchmark problems, it is understood that the algorithm successfully avoids the local optimum points of the problem in the parts that show an increase in the average distance curve of the MsRwGWO algorithm during the exploration phase. This is due to the mutation operator added to the GWO algorithm and the new gray wolf update mechanism.

4.2 Comparison between MsRwGWO and GWO for CEC2014 benchmark problems

To evaluate the performance of the proposed MsRwGWO algorithm, some numerical optimization problems as known CEC 2014 test suite were utilized. CEC 2014 test suite has thirty benchmark functions that are minimization problems. These benchmark functions are divided into four groups: unimodal (FN1-FN3), simple multimodal (FN4-FN16), hybrid (FN17-FN22), and composition functions (FN23-FN30). All optimization benchmark test problems with specific problem dimensions (10D, 30D, and 50D) were solved for 51 independent runs using the original GWO and proposed MsRwGWO algorithms. In the benchmark tests, we used the population size as 10 times the number of the dimension and the maximum iteration number as 1000. In solving optimization problems, we preferred to use the termination criterion as reaching the maximum number of iterations. The codes of GWO and MsRwGWO algorithms have been run on PC with Intel(R) Core(TM) i7-6500U CPU@2.50 GHz with 8 GB RAM. In solving CEC 2014 test problems, 14 error values were recorded for each function at each run. Figure 14 presents best convergence curves of some benchmarks with 10D for both algorithms. In only two of the benchmarks with different properties (F7 and F29), the GWO algorithm

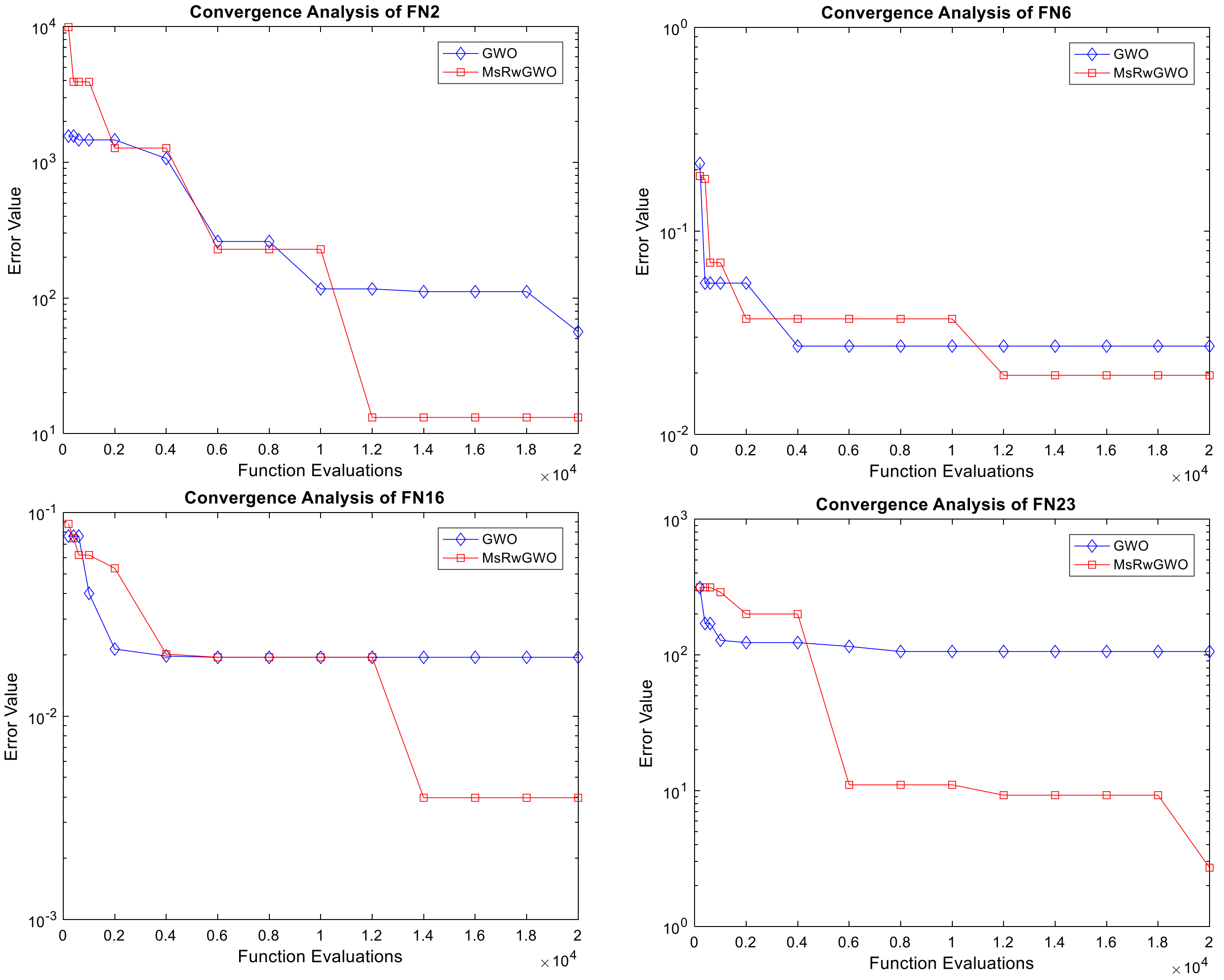


Fig. 7 Convergence analysis of MsRwGWO algorithm

could find a better result at the end of the optimization. It can be seen from these curves that the proposed MsRwGWO algorithm clearly has a better convergence. In Fig. 15, the best, worst, and mean convergence curves obtained by MsRwGWO algorithm at the end of the 51 runs are shown. These results show that the best and worst convergence curves of the MsRwGWO algorithm are close to the mean, that is, its standard deviation is low. This reveals that the proposed algorithm can solve the problems in a stable way. In the comparison results, we evaluated five metrics such as mean, worst, best, median, and standard deviation. In Tables 2, 3 and 4, the statistical results of GWO and MsRwGWO algorithms are given, respectively, for 10D, 30D, and 50D CEC2014 benchmark problems. In these tables, the best results among the metrics have been emphasized in boldface.

In the most of CEC2014 benchmarks, the proposed MsRwGWO in all problem dimensions has better performance than the original GWO in terms of all statistical metrics. To better show all these statistical results, a summary for all problem dimensions is provided in Table 5. As can be seen from this summary result table, MsRwGWO has 53.33% better results than GWO in 30 and 50 problem dimensions according to the best error metric. More interestingly, the success rate of the MsRwGWO algorithm for all dimensions (70% for 10D, 76.67% for 30D, and 70% for 50D) is much higher than GWO in the worst error metric. Mean error metric summary results show us that while the problem dimension increases, the proposed MsRwGWO algorithm has the same performance with GWO (both algorithms are the same for 50D). For 10D and 30D, the MsRwGWO provides 70% and 60% better results than the GWO algorithm, respectively. Finally, looking at the total of all metrics, it is understood that the MsRwGWO has a success rate of 64.67% for 10D, 70% for 30D, and 58% for 50D. In the last row of this table (total), the more successful of both algorithms is indicated in bold.

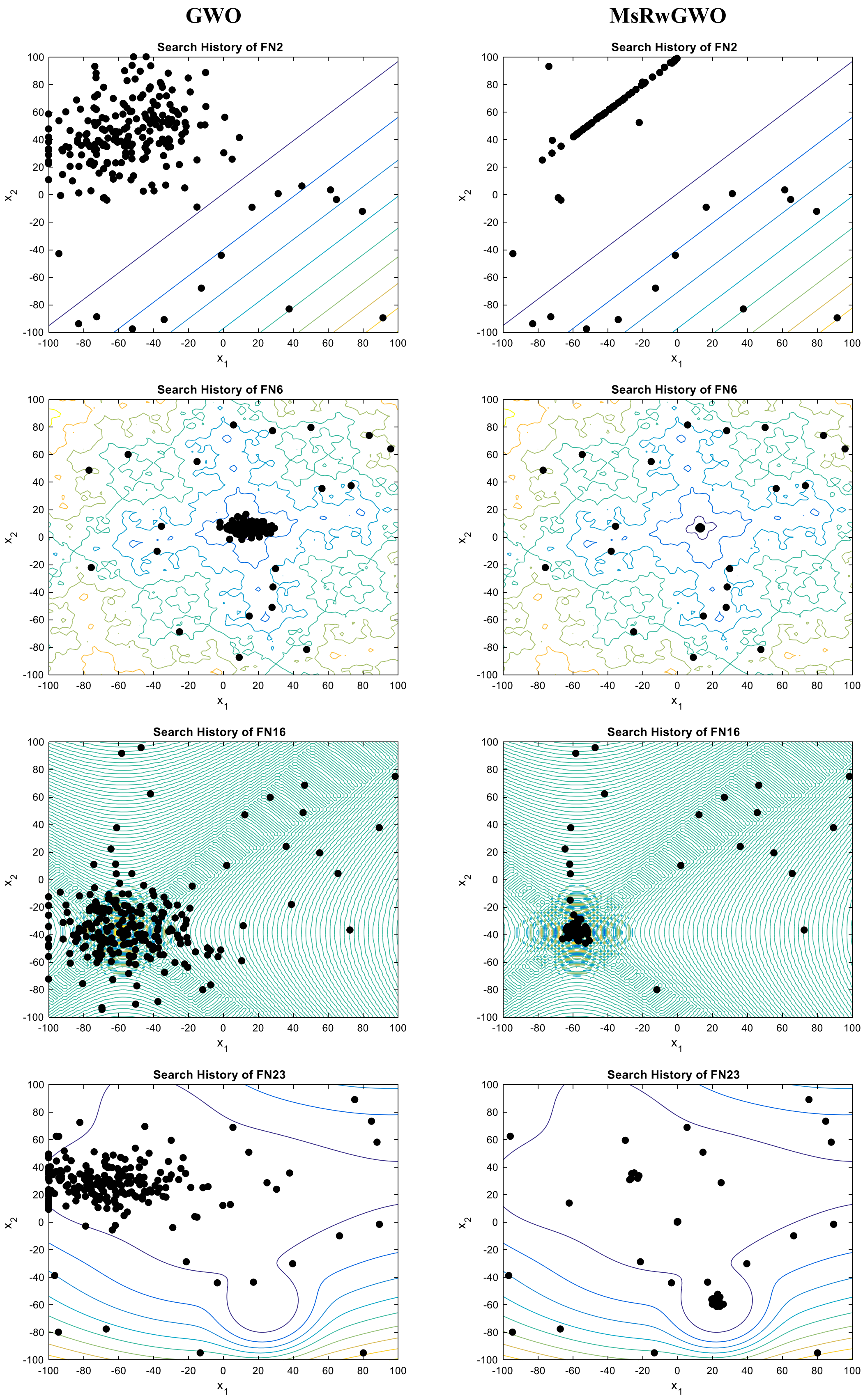


Fig. 8 Search history analysis of MsRwGWO algorithm

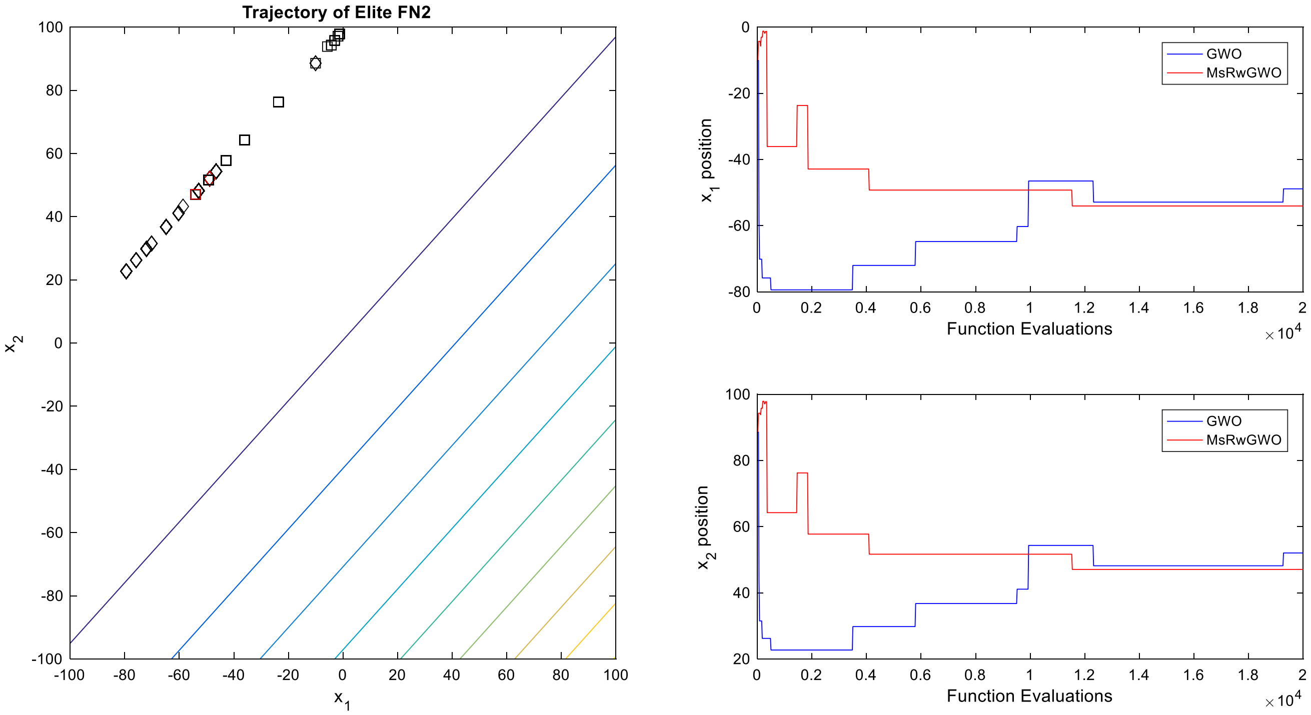


Fig. 9 Trajectory of alpha gray wolf for FN2 function (◇: GWO. □: MsRwGWO)

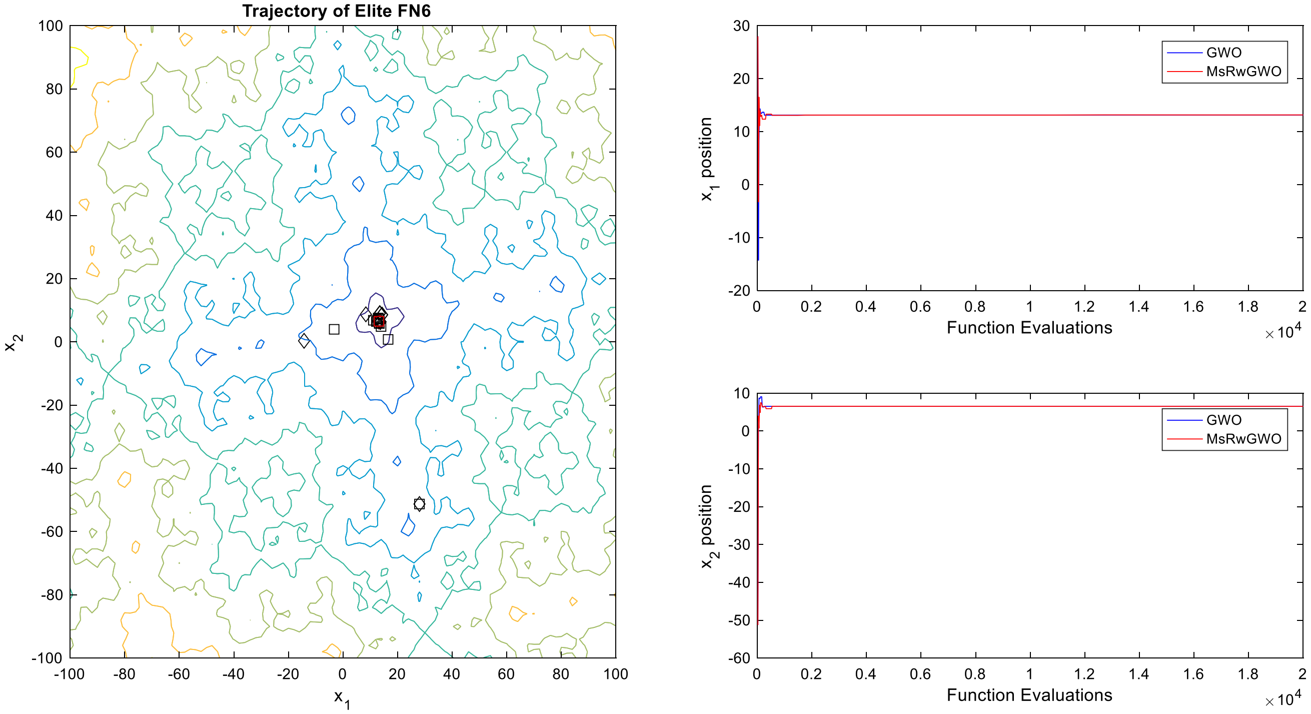


Fig. 10 Trajectory of alpha gray wolf for FN6 function (◇: GWO. □: MsRwGWO)

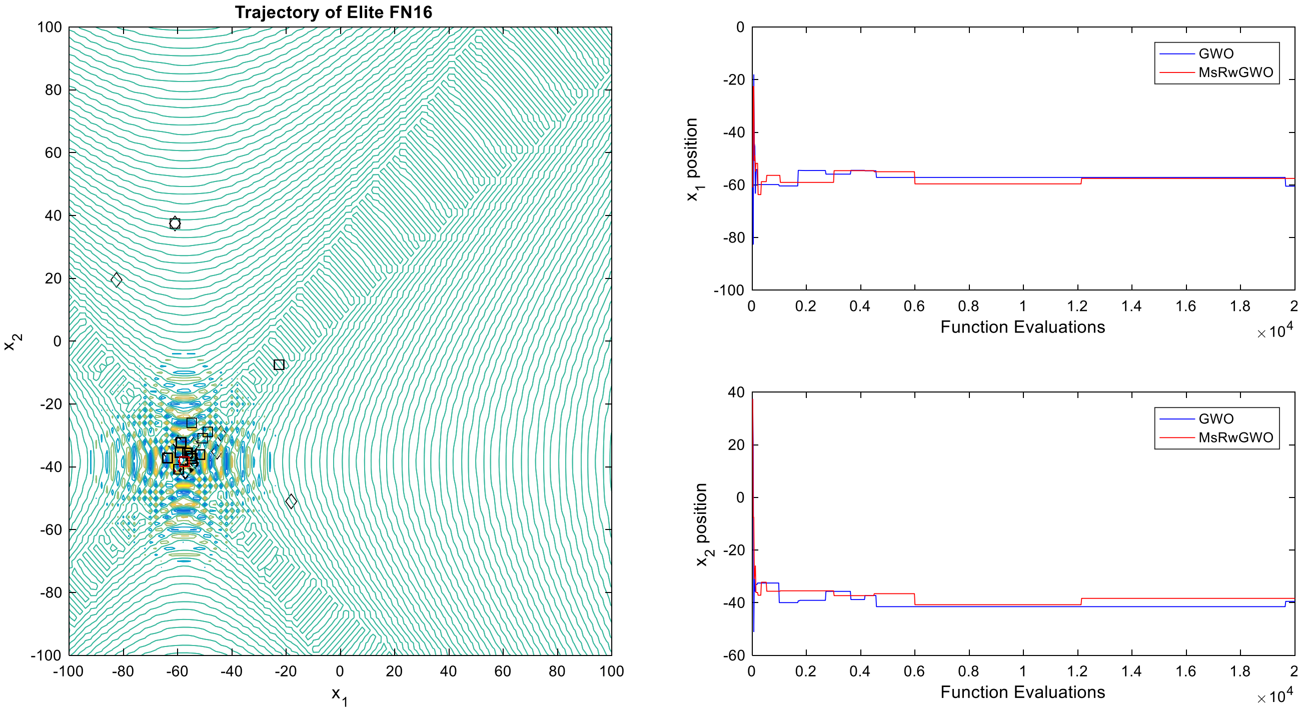


Fig. 11 Trajectory of alpha gray wolf for FN16 function (◇: GWO. □: MsRwGWO)

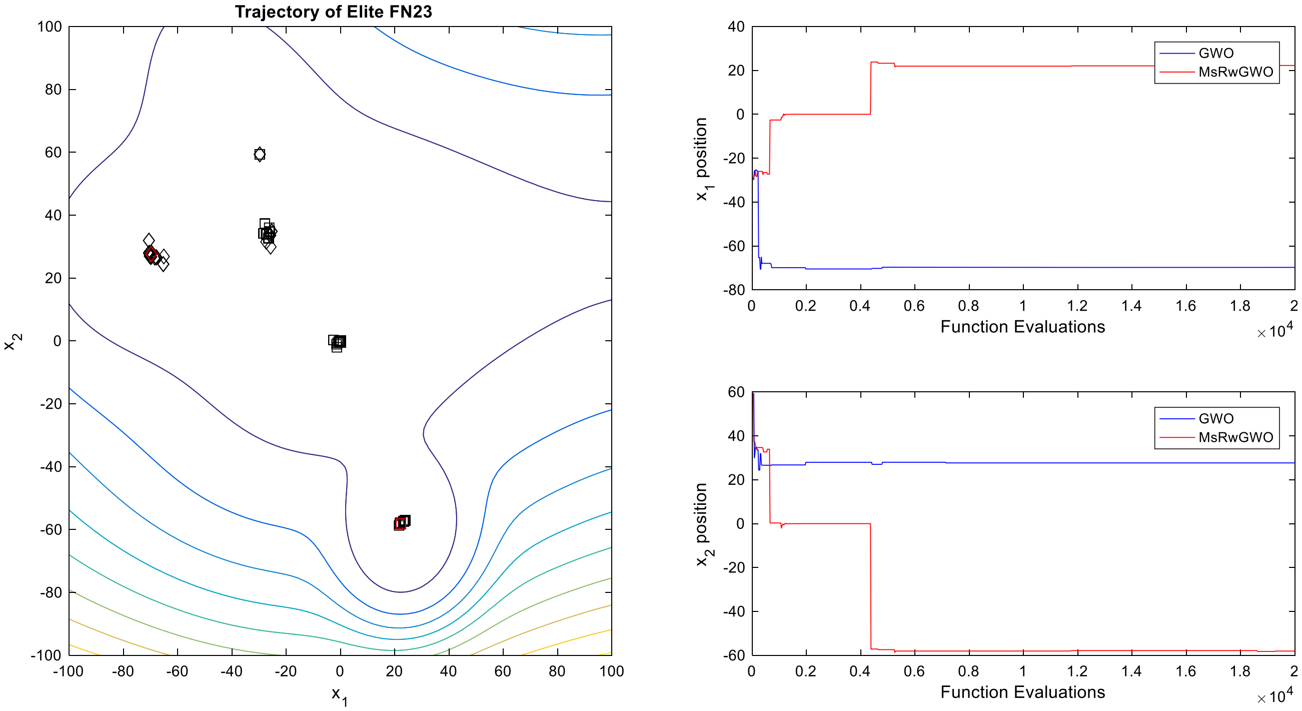


Fig. 12 Trajectory of alpha gray wolf for FN23 function (◇: GWO. □: MsRwGWO)

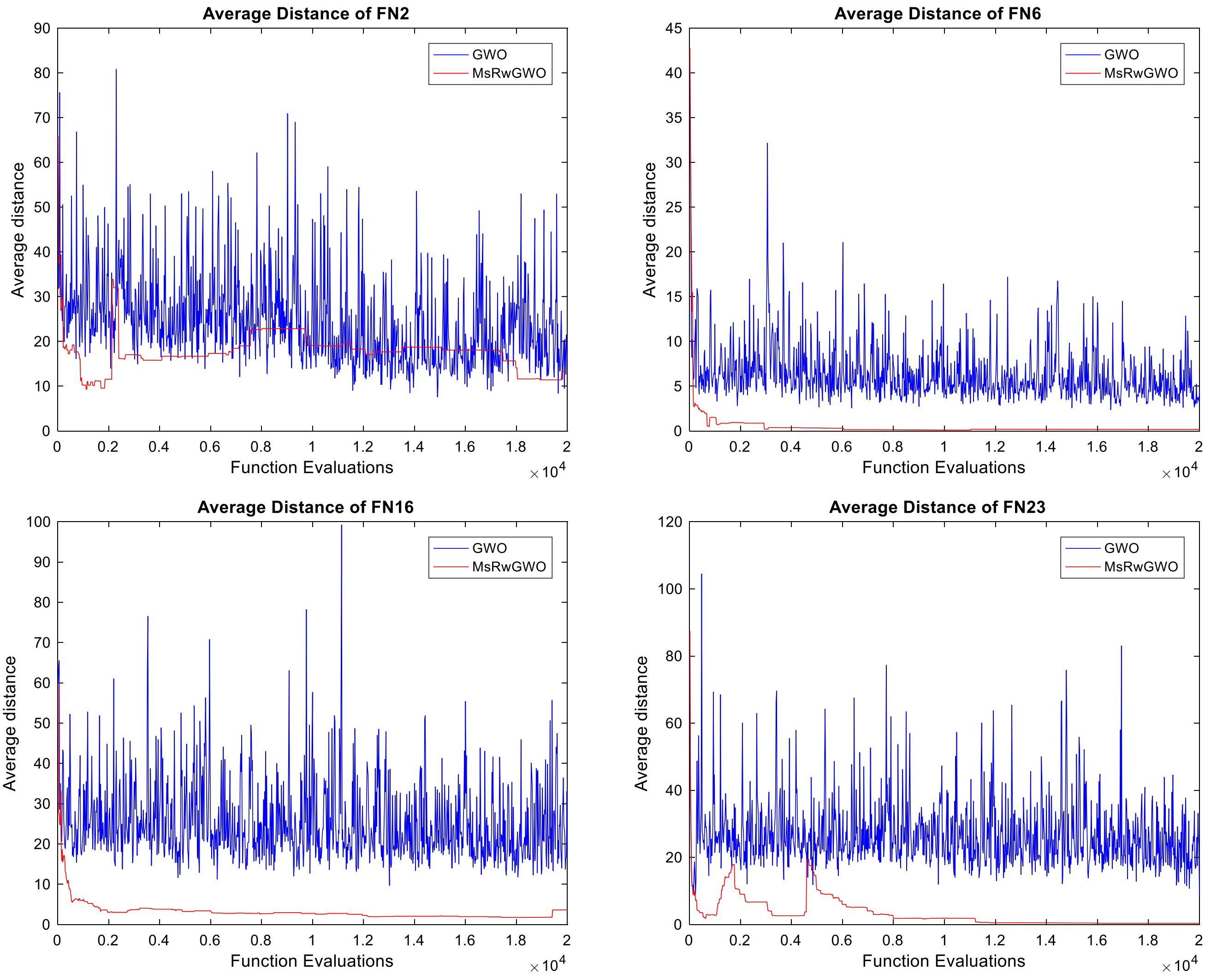


Fig. 13 Average distance analysis between gray wolves

4.3 Comparison of MsRwGWO with other algorithms

In addition to comparison between GWO and MsRwGWO, we also compared the proposed MsRwGWO algorithm with the popular and state-of-the-art meta-heuristic algorithms. In this comparison study, Moth-Flame Optimizer (MFO) [49], Particle Swarm Optimizer (PSO) [2], Dragonfly Algorithm (DA) [50], Sine Cosine Algorithm (SCA) [32], and Whale Optimization Algorithm (WOA) [52] were used. The parameters of the selected algorithms were set as in the original papers. In Table 6, the comparison results are presented for 30D CEC 2014 benchmark functions.

In this table, mean and standard deviation metrics are presented for all algorithms and they are ranked according to the mean error values of benchmark functions. From the average and overall ranks given at the end of Table 6, it is clear that the proposed MsRwGWO algorithm outperforms other meta-heuristic algorithms. As a result, the comparative results with CEC2014 benchmark functions used in this study show that different mechanisms such as transition mechanism, new weighted updating mechanism, novel checking boundary mechanism, renewed update mechanism of alpha, beta, and delta wolves, added into the MsRwGWO increase the performance of the algorithm in the exploration and exploitation.

Here, we have also compared the performance of the proposed MsRwGWO algorithm with those of some GWO algorithms taken from the literature. This comparison includes the variants of GWO such as improved GWO (IGWO) [78], opposition-based GWO (OBGWO) [75], and exploration-enhanced GWO (EEGWO) [79]. Table 7 summarizes the comparison results of the MsRwGWO algorithm and the other variants of GWO for 30-dimensional CEC2014 problems. This comparison was prepared

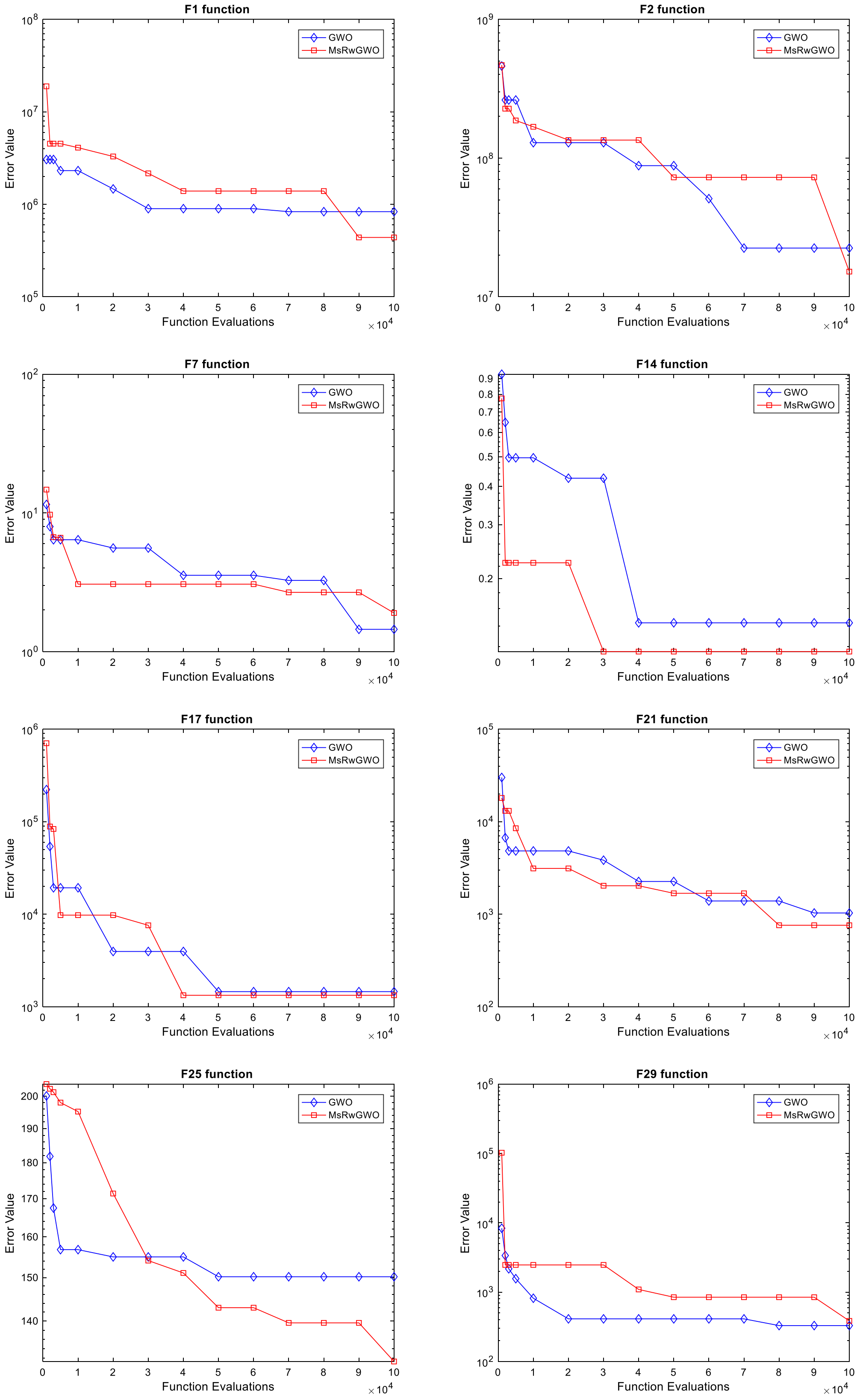


Fig. 14 Convergence curves of 10D best benchmark results for GWO and MsRwGWO

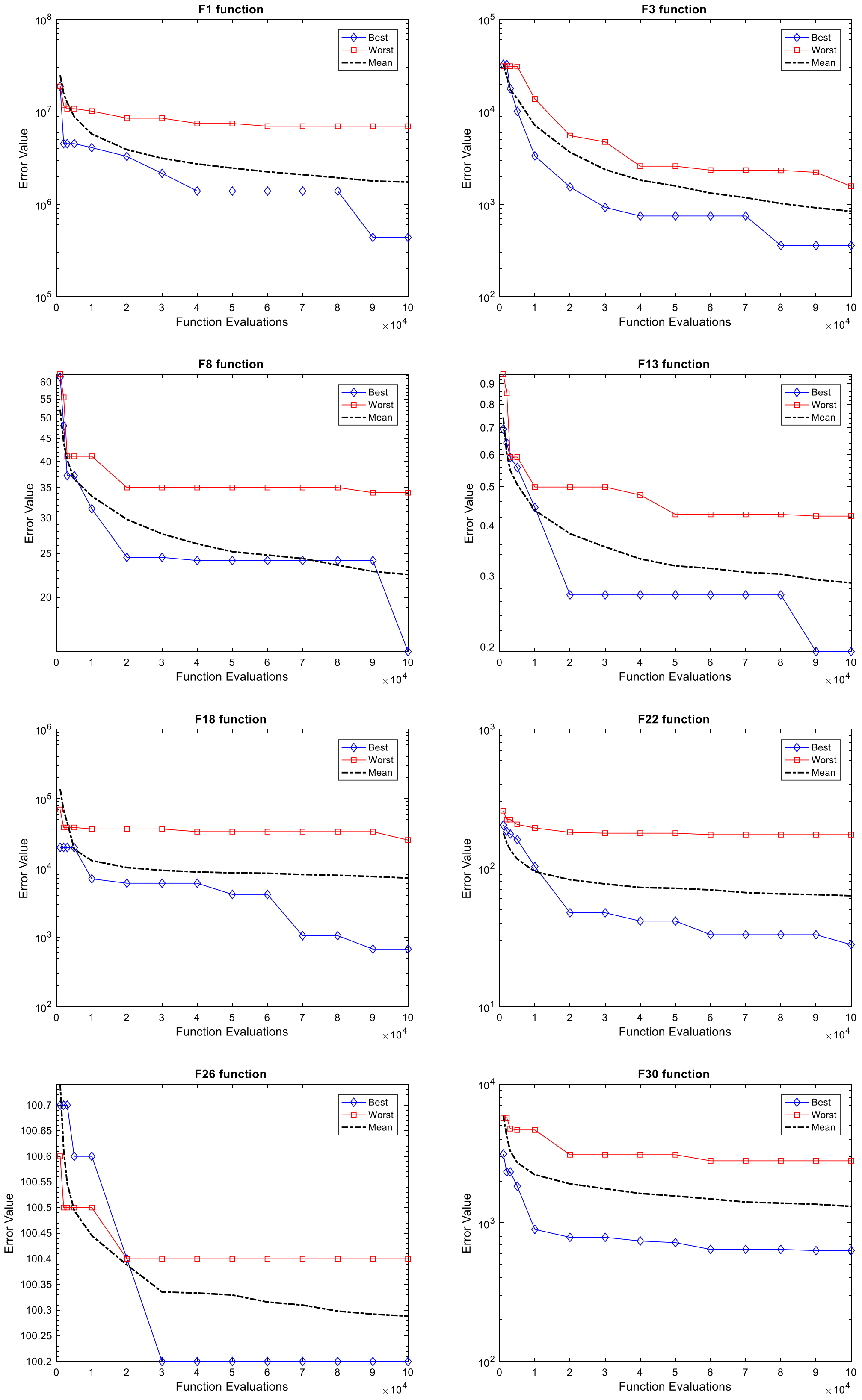


Fig. 15. 10D Convergence curves of MsRwGWO algorithm (best, worst, mean)

Table 2 10D CEC2014 benchmark results for GWO and MsRwGWO algorithms

No	GWO					MsRwGWO				
	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
1	8.32E+5	1.70E+7	4.29E+6	4.94E+6	3.39E+6	**4.38E+5**	**6.99E+6**	**1.27E+6**	**1.74E+6**	**1.28E+6**
2	2.03E+7	1.29E+8	7.06E+8	7.35E+8	2.19E+8	**1.52E+7**	**1.27E+8**	**7.74E+7**	**7.23E+7**	**2.26E+7**
3	5.51E+2	9.27E+3	1.84E+3	3.06E+3	2.43E+3	**3.58E+2**	**1.57E+3**	**8.18E+2**	**8.40E+2**	**2.55E+2**
4	1.07E+1	4.02E+1	**3.63E+1**	3.37E+1	**7.86E+0**	**9.57E+0**	**3.82E+1**	3.70E+1	**3.26E+1**	9.46E+0
5	**1.03E+1**	**2.05E+1**	2.04E+1	**1.97E+1**	2.19E+0	2.02E+1	2.05E+1	**2.04E+1**	2.04E+1	**7.99E−2**
6	2.29E+0	6.23E+0	3.40E+0	3.55E+0	8.31E−1	**2.21E+0**	**5.52E+0**	**3.19E+0**	**3.38E+0**	**7.39E−1**
7	**1.45E+0**	5.54E+0	3.04E+0	3.25E+0	9.26E−1	1.90E+0	**4.42E+0**	**2.84E+0**	**2.86E+0**	**4.93E−1**
8	**1.29E+1**	**3.26E+1**	**1.82E+1**	**1.92E+1**	4.21E+0	1.52E+1	3.41E+1	2.28E+1	2.25E+1	**3.79E+0**
9	**1.42E+1**	4.03E+1	**2.28E+1**	**2.30E+1**	4.57E+0	1.78E+1	**3.54E+1**	2.50E+1	2.48E+1	**3.98E+0**
10	1.66E+2	**8.21E+2**	**3.72E+2**	**4.04E+2**	**1.43E+2**	**1.12E+2**	1.01E+3	4.22E+2	4.53E+2	2.00E+2
11	**2.39E+2**	**1.25E+3**	**7.10E+2**	**7.18E+2**	2.50E+2	4.01E+2	1.50E+3	7.59E+2	7.93E+2	**2.42E+2**
12	**5.59E−1**	**1.53E+0**	**1.13E+0**	**1.08E+0**	2.34E−1	5.97E−1	1.63E+0	1.21E+0	1.17E+0	**2.13E−1**
13	**1.88E−−1**	4.70E−1	3.27E−1	3.16E−1	6.78E−2	1.95E−1	**4.23E−1**	**2.88E−1**	**2.89E−1**	**5.03E−2**
14	1.44E−1	8.51E−1	2.64E−1	3.27E−1	1.92E−1	**1.16E−1**	**3.26E−1**	**2.23E−1**	**2.22E−1**	**4.92E−2**
15	2.10E+0	5.65E+0	**3.36E+0**	3.43E+0	6.82E−1	**1.98E+0**	**4.39E+0**	3.44E+0	**3.35E+0**	**5.76E−1**
16	**1.93E+0**	3.41E+0	**2.55E+0**	2.62E+0	3.05E−1	1.93E+0	**3.23E+0**	2.60E+0	**2.60E+0**	**2.73E−1**
17	1.45E+3	2.69E+4	5.48E+3	7.20E+3	4.78E+3	**1.33E+3**	**1.09E+4**	**4.28E+3**	**5.18E+3**	**2.48E+3**
18	**1.56E+2**	2.55E+4	6.46E+3	8.59E+3	6.94E+3	6.76E+2	**2.52E+4**	**5.90E+3**	**7.16E+3**	**5.63E+3**
19	2.12E+0	5.59E+0	3.11E+0	3.34E+0	7.68E−1	**1.99E+0**	**5.12E+0**	**2.96E+0**	**3.07E+0**	**5.90E−1**
20	**3.90E+1**	1.12E+4	2.06E+3	3.11E+3	3.55E+3	7.31E+1	**4.04E+2**	**1.61E+2**	**1.71E+2**	**7.41E+1**
21	1.03E+3	**1.22E+4**	5.29E+3	5.93E+3	3.80E+3	**7.60E+2**	1.29E+4	**2.61E+3**	**4.01E+3**	**3.16E+3**
22	**2.51E+1**	**1.71E+2**	**4.52E+1**	**6.20E+1**	**4.56E+1**	2.80E+1	1.74E+2	4.53E+1	6.30E+1	4.68E+1
23	**3.30E+2**	3.37E+2	3.31E+2	3.32E+2	1.94E+0	**3.30E+2**	**3.32E+2**	**3.31E+2**	**3.31E+2**	**4.67E−1**
24	1.22E+2	2.04E+2	**1.31E+2**	1.36E+2	1.77E+1	**1.20E+2**	**2.03E+2**	1.32E+2	**1.33E+2**	**1.15E+1**
25	1.50E+2	2.03E+2	2.01E+2	2.00E+2	**7.32E+0**	**1.31E+2**	**2.02E+2**	**2.00E+2**	**1.82E+2**	2.63E+1
26	**1.00E+2**	**1.00E+2**	**1.00E+2**	**1.00E+2**	**5.03E−2**	**1.00E+2**	**1.00E+2**	**1.00E+2**	**1.00E+2**	5.53E−2
27	**2.82E+0**	4.31E+2	3.92E+2	3.35E+2	**1.44E+2**	4.11E+0	**4.11E+2**	**3.91E+2**	**3.11E+2**	1.62E+2
28	3.87E+2	6.02E+2	4.64E+2	4.56E+2	5.66E+1	**3.84E+2**	**5.22E+2**	**4.54E+2**	**4.52E+2**	**5.07E+1**
29	**3.41E+2**	**2.13E+6**	**8.40E+2**	2.30E+5	6.36E+5	3.85E+2	2.41E+6	9.30E+2	**1.25E+5**	**5.05E+5**
30	**5.07E+2**	2.80E+3	**1.01E+3**	**1.19E+3**	**5.38E+2**	6.30E+2	**2.80E+3**	1.23E+3	1.31E+3	5.50E+2

according to the mean of the errors of the objective function values obtained as a result of repeated running. The ranking results of all algorithms for each benchmark function are given in the table, and the average ranking result of the proposed MsRwGWO algorithm and GWO variants for all benchmarks is presented in the last row of the table. This ranking result clearly indicates the superiority of the MsRwGWO algorithm against the other GWO variants.

4.4 Short-term wind speed forecasting using MsRwGWO-MLP hybrid model

In this section, the proposed MsRwGWO algorithm was adapted to a Multi-Layer Perceptron (MLP) model for wind speed estimation as a real-world application. For the planning and management of power systems, it is of great importance to determine the electrical energy to be obtained from wind energy in the next horizon steps. Due to the chaotic and uncertain structure of wind speed, different models are proposed by researchers today to increase the performance of short-term forecasting researches [80–84]. Nowadays, hybrid models with meta-heuristic approaches have become popular in this field of research.

The MsRwGWO is utilized to optimize the parameters of the MLP model in its training phase. In the MsRwGWO-MLP hybrid model, all gray wolves are encoded as one-dimensional vectors of randomly generated real values in range [− 10, 10]. This encoding vector consists of two parts: connection weights among the layers and bias values of hidden and output layers. In the optimization of weight and bias values of the MLP model, the problem dimension

Table 3 30D CEC2014 benchmark results for GWO and MsRwGWO algorithms

No	GWO					MsRwGWO				
	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
1	**2.61E+7**	1.16E+8	5.68E+7	5.65E+7	2.12E+7	2.95E+7	**9.09E+7**	**5.26E+7**	**5.53E+7**	**1.27E+7**
2	**2.09E+9**	4.55E+9	3.02E+9	3.13E+9	5.55E+8	2.27E+9	**4.13E+9**	**2.94E+9**	**2.97E+9**	**3.63E+8**
3	9.26E+3	3.99E+4	2.20E+4	2.14E+4	5.94E+3	**6.29E+3**	**1.29E+4**	**9.21E+3**	**9.30E+3**	**1.67E+3**
4	**1.83E+2**	4.88E+2	2.59E+2	2.65E+2	4.95E+1	2.05E+2	**3.13E+2**	**2.54E+2**	**2.56E+2**	**2.27E+1**
5	2.08E+1	**2.10E+1**	**2.10E+1**	**2.09E+1**	5.43E−2	**2.08E+1**	2.11E+1	**2.10E+1**	2.10E+1	**5.39E−2**
6	1.80E+1	2.73E+1	2.17E+1	2.21E+1	1.81E+0	**1.78E+1**	**2.44E+1**	**2.03E+1**	**2.03E+1**	**1.29E+0**
7	2.11E+1	6.64E+1	2.96E+1	3.01E+1	6.58E+0	**1.84E+1**	**3.34E+1**	**2.59E+1**	**2.57E+1**	**3.19E+0**
8	**1.08E+2**	**2.00E+2**	**1.34E+2**	**1.36E+2**	1.52E+1	1.36E+2	2.17E+2	1.68E+2	1.68E+2	**1.50E+1**
9	**1.41E+2**	**2.19E+2**	**1.95E+2**	**1.92E+2**	1.57E+1	1.63E+2	2.24E+2	1.98E+2	1.97E+2	**1.17E+1**
10	**2.51E+3**	**4.64E+3**	**3.51E+3**	**3.51E+3**	**5.10E+2**	3.47E+3	6.08E+3	4.85E+3	4.74E+3	5.79E+2
11	**4.14E+3**	**6.89E+3**	**5.91E+3**	**5.79E+3**	6.90E+2	5.54E+3	7.20E+3	6.60E+3	6.48E+3	**3.76E+2**
12	**1.65E+0**	2.89E+0	2.50E+0	**2.40E+0**	3.04E−1	1.78E+0	**2.87E+0**	**2.45E+0**	2.46E+0	**2.35E−1**
13	**5.43E−1**	8.96E−1	6.96E−1	6.99E−1	6.90E−2	5.44E−1	**8.38E−1**	**6.40E−1**	**6.53E−1**	**6.57E−2**
14	1.83E+0	1.55E+1	6.47E+0	6.67E+0	2.93E+0	**1.71E+0**	**8.88E+0**	**4.26E+0**	**4.52E+0**	**1.98E+0**
15	2.94E+1	1.40E+2	3.93E+1	4.18E+1	1.53E+1	**2.92E+1**	**6.35E+1**	**3.55E+1**	**3.68E+1**	**5.51E+0**
16	**1.09E+1**	**1.25E+1**	**1.18E+1**	**1.17E+1**	3.17E−1	1.12E+1	**1.25E+1**	1.19E+1	1.19E+1	**3.06E−1**
17	4.50E+5	5.88E+6	**1.16E+6**	**1.40E+6**	9.51E+5	**3.68E+5**	**3.85E+6**	1.31E+6	1.46E+6	**6.89E+5**
18	8.36E+6	7.09E+7	2.23E+7	2.57E+7	1.54E+7	**4.70E+6**	**5.68E+7**	**2.18E+7**	**2.44E+7**	**1.19E+7**
19	2.24E+1	7.67E+1	2.87E+1	3.09E+1	8.73E+0	**2.21E+1**	**3.57E+1**	**2.76E+1**	**2.79E+1**	**3.21E+0**
20	1.60E+3	1.83E+4	9.71E+3	9.83E+3	4.39E+3	**8.23E+2**	**3.50E+3**	**1.66E+3**	**1.76E+3**	**6.16E+2**
21	1.33E+5	1.66E+6	3.69E+5	5.12E+5	3.89E+5	**1.16E+5**	**1.12E+6**	**3.22E+5**	**4.00E+5**	**2.31E+5**
22	2.49E+2	7.78E+2	**3.68E+2**	**4.11E+2**	1.34E+2	**2.17E+2**	**7.23E+2**	4.72E+2	4.52E+2	**1.31E+2**
23	3.24E+2	3.57E+2	3.30E+2	3.31E+2	4.90E+0	**3.22E+2**	**3.31E+2**	**3.26E+2**	**3.26E+2**	**2.03E+0**
24	**2.00E+2**	**2.00E+2**	**2.00E+2**	**2.00E+2**	**0.00E+0**	**2.00E+2**	**2.00E+2**	**2.00E+2**	**2.00E+2**	**0.00E+0**
25	2.06E+2	2.17E+2	2.12E+2	2.11E+2	2.56E+0	**2.00E+2**	**2.14E+2**	**2.10E+2**	**2.10E+2**	**2.41E+0**
26	**1.01E+2**	**1.01E+2**	**1.01E+2**	**1.01E+2**	**8.20E−2**	**1.01E+2**	2.01E+2	**1.01E+2**	1.10E+2	3.00E+1
27	**4.21E+2**	9.81E+2	8.54E+2	7.78E+2	1.89E+2	4.23E+2	**9.37E+2**	**7.97E+2**	**7.23E+2**	**1.85E+2**
28	**1.04E+3**	2.27E+3	**1.17E+3**	1.25E+3	2.38E+2	1.13E+3	**1.56E+3**	1.22E+3	**1.24E+3**	**7.46E+1**
29	**2.00E+5**	**1.31E+6**	**4.87E+5**	**5.56E+5**	**2.82E+5**	3.10E+5	2.00E+6	7.70E+5	9.39E+5	4.99E+5
30	**1.37E+4**	1.48E+5	4.14E+4	**4.35E+4**	2.22E+4	1.67E+4	**8.62E+4**	**3.85E+4**	4.37E+4	**1.71E+4**

is the length of this vector and it can be calculated as given below:

$$D = N_I \times N_H + N_H \times N_O + N_H + N_O \tag{18}$$

where N_I denotes the number of inputs, N_H stands for the number of the neurons in the hidden layer, and N_O represents the number of outputs. Figure 16 shows how MLP model parameters are optimized by the MsRwGWO algorithm.

The wind speed datasets used in this paper are collected from a wind farm in Balıkesir, Turkey. Each series contains 6849 samples and is divided into training series and testing series. The first 4794 samples of each site series are used for training, and the rest are used for testing. The height of the measured wind speed is 50 m, and the sampling interval is 15 min. In order to increase the model performance, the input dataset is normalized in range of [0 1]. For the one-step short-term wind speed forecasting, we used three sequential inputs $(V(k), V(k-1), V(k-2))$ of the wind speed dataset in the MLP model based on MsRwGWO.

In optimizing the parameters of the MLP model, the objective function was used as the RMSE function for the examples in the training dataset. The MsRwGWO used in the optimization of MLP model parameters is run 50 times independently, and the performance results are calculated statistically for training and test phases of MsRwGWO-MLP model. Training and test results of the MLP model with the best parameters optimized by MsRwGWO

Table 4 50D CEC2014 benchmark results for GWO and MsRwGWO algorithms

No	GWO					MsRwGWO				
	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
1	**6.90E+7**	2.33E+8	**1.23E+8**	**1.29E+8**	**2.96E+7**	8.36E+7	**2.27E+8**	1.30E+8	1.36E+8	3.02E+7
2	7.83E+9	1.57E+10	1.07E+10	1.10E+10	1.86E+9	**6.64E+9**	**1.22E+10**	**9.63E+9**	**9.66E+9**	**1.03E+9**
3	2.01E+4	7.61E+4	4.17E+4	4.16E+4	1.07E+4	**1.80E+4**	**3.18E+4**	**2.40E+4**	**2.42E+4**	**3.23E+3**
4	**5.51E+2**	1.77E+3	8.25E+2	8.59E+2	2.06E+2	5.57E+2	**1.13E+3**	**7.44E+2**	**7.65E+2**	**1.14E+2**
5	2.11E+1	2.12E+1	2.11E+1	2.11E+1	**3.29E−2**	**2.10E+1**	**2.12E+1**	**2.11E+1**	**2.11E+1**	4.65E−2
6	3.75E+1	4.97E+1	4.35E+1	4.37E+1	2.57E+0	**3.54E+1**	**4.71E+1**	**4.14E+1**	**4.16E+1**	**2.53E+0**
7	6.60E+1	1.32E+2	8.71E+1	8.93E+1	1.32E+1	**6.34E+1**	**1.07E+2**	**7.94E+1**	**8.11E+1**	**9.00E+0**
8	**2.60E+2**	**3.89E+2**	**3.24E+2**	**3.26E+2**	2.39E+1	3.30E+2	4.17E+2	3.75E+2	3.75E+2	**2.17E+1**
9	3.69E+2	**4.39E+2**	4.04E+2	4.04E+2	**1.91E+1**	**3.50E+2**	4.50E+2	**4.03E+2**	**4.03E+2**	1.96E+1
10	**6.97E+3**	**1.03E+4**	**8.71E+3**	**8.79E+3**	**7.07E+2**	8.24E+3	1.27E+4	1.10E+4	1.10E+4	9.74E+2
11	**1.05E+4**	**1.32E+4**	**1.21E+4**	**1.20E+4**	6.76E+2	1.18E+4	1.35E+4	1.28E+4	1.28E+4	**4.26E+2**
12	**2.19E+0**	3.86E+0	**3.28E+0**	**3.26E+0**	3.12E−1	2.61E+0	**3.82E+0**	3.34E+0	3.28E+0	**2.85E−1**
13	7.52E−1	1.37E+0	9.79E−1	9.82E−1	1.11E−1	**7.39E−1**	**1.17E+0**	**9.20E−1**	**9.14E−1**	**7.79E−2**
14	1.80E+1	3.91E+1	2.41E+1	2.44E+1	4.19E+0	**1.65E+1**	**2.64E+1**	**2.08E+1**	**2.11E+1**	**2.12E+0**
15	**1.17E+2**	3.08E+3	5.07E+2	6.89E+2	5.86E+2	1.45E+2	**6.09E+2**	**2.87E+2**	**3.06E+2**	**1.03E+2**
16	**2.03E+1**	**2.20E+1**	**2.12E+1**	**2.12E+1**	4.12E−1	2.06E+1	2.20E+1	2.14E+1	2.14E+1	**3.21E−1**
17	**2.34E+6**	1.30E+7	**6.83E+6**	**7.03E+6**	2.79E+6	3.51E+6	**1.28E+7**	7.50E+6	7.43E+6	**2.30E+6**
18	1.24E+8	3.80E+8	**2.19E+8**	**2.16E+8**	**4.56E+7**	**9.43E+7**	**3.61E+8**	2.24E+8	2.24E+8	5.65E+7
19	7.02E+1	1.14E+2	9.72E+1	9.54E+1	**1.07E+1**	**6.68E+1**	**1.12E+2**	**9.05E+1**	**9.05E+1**	1.42E+1
20	2.20E+3	2.40E+4	1.07E+4	1.14E+4	4.81E+3	**1.97E+3**	**6.93E+3**	**3.44E+3**	**3.52E+3**	**9.01E+2**
21	1.00E+6	**5.95E+6**	2.46E+6	2.59E+6	**1.11E+6**	**9.27E+5**	6.66E+6	**2.11E+6**	**2.46E+6**	1.15E+6
22	**6.20E+2**	2.08E+3	**1.32E+3**	**1.33E+3**	3.02E+2	9.92E+2	**1.86E+3**	1.48E+3	1.48E+3	**2.06E+2**
23	4.05E+2	4.92E+2	4.38E+2	4.44E+2	1.90E+1	**3.94E+2**	**4.55E+2**	**4.25E+2**	**4.24E+2**	**1.22E+1**
24	**2.00E+2**	**2.00E+2**	**2.00E+2**	**2.00E+2**	**0.00E+0**	**2.00E+2**	**2.00E+2**	**2.00E+2**	2.00E+2	5.05E−2
25	**2.19E+2**	2.43E+2	**2.29E+2**	**2.29E+2**	4.94E+0	**2.19E+2**	**2.39E+2**	2.31E+2	2.30E+2	**4.03E+0**
26	**1.01E+2**	3.91E+2	**2.00E+2**	**1.56E+2**	6.05E+1	1.01E+2	**2.10E+2**	2.03E+2	1.76E+2	**4.67E+1**
27	1.32E+3	1.64E+3	1.45E+3	1.45E+3	**6.00E+1**	**1.24E+3**	**1.54E+3**	**1.40E+3**	**1.40E+3**	6.80E+1
28	**1.90E+3**	**4.27E+3**	**2.27E+3**	**2.49E+3**	6.04E+2	2.16E+3	4.58E+3	2.44E+3	2.57E+3	**4.01E+2**
29	**4.00E+6**	2.93E+7	**7.90E+6**	**8.93E+6**	**4.12E+6**	6.30E+6	**2.72E+7**	1.41E+7	1.44E+7	4.89E + 6
30	**8.69E+4**	**3.71E+5**	**2.04E+5**	**2.07E+5**	**6.85E+4**	1.45E+5	5.80E+5	2.56E+5	2.74E+5	8.29E+4

Table 5 Summary results of CEC 2014 benchmarks for GWO and MsRwGWO algorithms

	GWO			MsRwGWO		
	10D (%)	30D (%)	50D (%)	10D (%)	30D (%)	50D (%)
Best	50.00	46.67	46.67	50.00	53.33	53.33
Worst	30.00	23.33	30.00	70.00	76.67	70.00
Median	43.33	30.00	46.67	56.67	70.00	53.33
Mean	30.00	40.00	50.00	70.00	60.00	50.00
Std	23.33	10.00	36.67	76.67	90.00	63.33
Total	35.33	30.00	42.00	**64.67**	**70.00**	**58.00**

Table 6 Comparison results of MsRwGWO and other algorithms for 30D CEC2014 problems

Nos.		MFO	PSO	DA	SCA	WOA	MsRwGWO
1	Mean	7.59E+7	8.19E+7	4.49E+8	2.24E+8	3.24E+7	5.53E+7
	Std	9.77E+7	8.27E+9	3.88E+8	7.18E+7	1.42E+7	1.27E+7
	Rank	3	4	6	5	1	2
2	Mean	1.36E+10	2.81E+4	3.67E+10	1.60E+10	5.05E+6	2.97E+9
	Std	8.42E+9	5.30E+2	1.55E+10	3.12E+9	4.79E+6	3.63E+8
	Rank	4	1	6	5	2	3
3	Mean	8.99E+4	2.09E+1	1.81E+5	3.79E+4	2.97E+4	9.30E+3
	Std	4.98E+4	2.77E+1	9.61E+4	7.28E+3	2.10E+4	1.67E+3
	Rank	5	1	6	4	3	2
4	Mean	1.14E+3	9.01E+3	4.86E+3	1.02E+3	1.76E+2	2.56E+2
	Std	1.13E+3	1.85E+2	3.72E+3	2.04E+2	4.70E+1	2.27E+1
	Rank	4	6	5	3	1	2
5	Mean	2.04E+1	2.19E+2	2.08E+1	2.09E+1	2.04E+1	2.10E+1
	Std	1.75E−1	6.01E+3	1.17E−1	5.00E−2	1.80E−1	5.39E−2
	Rank	1	5	2	3	1	4
6	Mean	2.40E+1	6.78E+3	3.72E+1	3.44E+1	3.48E+1	2.03E+1
	Std	3.33E+0	2.46E+0	3.71E+0	2.61E+0	3.61E+0	1.29E+0
	Rank	2	6	5	3	4	1
7	Mean	1.17E+2	1.61E+0	3.54E+2	1.33E+2	1.03E+0	2.57E+1
	Std	6.91E+1	2.67E+1	1.59E+2	2.91E+1	5.00E−2	3.19E+0
	Rank	4	2	6	5	1	3
8	Mean	1.43E+2	1.10E+3	2.89E+2	2.36E+2	1.76E+2	1.68E+2
	Std	3.81E+1	1.24E+1	4.77E+2	2.08E+1	3.73E+1	1.50E+1
	Rank	1	6	5	4	3	2
9	Mean	2.23E+2	2.30E+6	3.01E+2	2.67E+2	2.13E+2	1.97E+2
	Std	6.06E+1	8.06E+7	6.73E+1	2.04E+1	5.22E+1	1.17E+1
	Rank	3	6	5	4	2	1
10	Mean	3.47E+3	4.43E+1	6.57E+3	5.88E+3	3.77E+3	4.74E+3
	Std	8.85E+2	7.82E+3	6.79E+2	4.51E+2	5.30E+2	5.79E+2
	Rank	2	1	6	5	3	4
11	Mean	4.15E+3	6.50E+5	6.92E+3	7.04E+3	4.50E+3	6.48E+3
	Std	6.90E+2	6.35E+2	7.41E+2	2.77E+2	7.58E+2	3.76E+2
	Rank	1	6	4	5	2	3
12	Mean	4.30E−1	3.73E+2	2.21E+0	2.45E+0	1.67E+0	2.46E+0
	Std	2.60E−1	2.76E+2	5.26E−1	2.30E-1	3.60E−1	2.35E−1
	Rank	1	6	3	4	2	5
13	Mean	2.21E+0	2.14E+2	5.14E+0	2.89E+0	5.00E−1	6.53E−1
	Std	1.34E+0	1.58E+2	1.65E+0	3.30E−1	1.20E−1	6.57E−2
	Rank	3	6	5	4	1	2
14	Mean	3.54E+1	9.05E+2	1.31E+2	4.11E+1	2.80E−1	4.52E+0
	Std	2.47E+1	1.92E+3	5.52E+1	5.51E+0	4.00E−2	1.98E+0
	Rank	3	6	5	4	1	2
15	Mean	2.23E+5	1.04E+7	2.85E+5	2.82E+3	7.00E+1	3.68E+1
	Std	5.77E+5	6.54E+4	4.25E+5	3.62E+3	2.52E+1	5.51E+0
	Rank	4	6	5	3	2	1
16	Mean	1.27E+1	2.49E+7	1.32E+1	1.28E+1	1.26E+1	1.19E+1
	Std	5.30E−1	2.21E+9	3.23E−1	3.10E−1	5.80E−1	3.06E−1
	Rank	3	6	5	4	2	1

Table 6 (continued)

Nos.		MFO	PSO	DA	SCA	WOA	MsRwGWO
17	Mean	3.39E+6	1.26E+4	1.61E+7	6.61E+6	4.33E+6	1.46E+6
	Std	4.07E+6	3.30E+2	1.36E+7	2.94E+6	2.22E+6	6.89E+5
	Rank	3	1	6	5	4	2
18	Mean	5.19E+6	4.80E−2	6.11E+8	1.85E+8	1.55E+4	2.44E+7
	Std	3.61E+7	1.43E+0	8.31E+8	8.68E+7	4.80E+4	1.19E+7
	Rank	3	1	6	5	2	4
19	Mean	7.36E+1	2.91E+1	2.68E+2	9.08E+1	4.48E+1	2.79E+1
	Std	5.32E+1	1.48E+1	1.08E+2	2.42E+1	3.14E+1	3.21E+0
	Rank	4	2	6	5	3	1
20	Mean	5.67E+4	1.34E+1	2.78E+5	1.31E+4	2.04E+4	1.76E+3
	Std	4.34E+4	4.16E+2	5.77E+5	4.83E+3	1.06E+4	6.16E+2
	Rank	5	1	6	3	4	2
21	Mean	7.83E+5	3.88E+2	8.23E+6	1.48E+6	9.48E+5	4.00E+5
	Std	1.18E+6	2.70E−1	8.77E+6	8.55E+5	9.91E+5	2.31E+5
	Rank	3	1	6	5	4	2
22	Mean	8.67E+4	7.40E−1	1.12E+3	7.54E+2	7.49E+2	4.52E+2
	Std	2.29E+4	1.19E+1	4.04E+2	1.25E+2	1.99E+2	1.31E+2
	Rank	6	1	5	4	3	2
23	Mean	3.71E+2	2.94E+3	5.76E+2	3.70E+2	3.31E+2	3.26E+2
	Std	3.98E+1	3.10E−1	1.98E+2	1.37E+1	6.24E+0	2.03E+0
	Rank	4	6	5	3	2	1
24	Mean	2.76E+2	1.05E+6	2.81E+2	2.01E+2	2.06E+2	2.00E+2
	Std	2.73E+1	4.44E+7	4.14E+1	9.10E−1	5.95E+0	0.00E+0
	Rank	4	6	5	2	3	1
25	Mean	2.14E+2	1.64E+1	2.40E+2	2.27E+2	2.25E+2	2.10E+2
	Std	7.65E+0	1.52E+4	2.28E+1	6.18E+0	1.66E+1	2.41E+0
	Rank	3	1	6	5	4	2
26	Mean	1.03E+2	2.83E+5	1.04E+2	1.02E+2	1.00E+2	1.10E+2
	Std	1.50E+0	1.99E+2	1.69E+0	5.60E−1	1.00E−1	3.00E+1
	Rank	3	6	4	2	1	5
27	Mean	9.21E+2	2.28E+1	1.09E+3	7.22E+2	9.05E+2	7.23E+2
	Std	2.23E+2	2.15E+0	3.58E+2	2.91E+2	4.15E+2	1.85E+2
	Rank	5	1	6	2	4	3
28	Mean	1.12E+3	2.59E+0	2.24E+3	2.00E+3	2.15E+3	1.24E+3
	Std	1.57E+2	5.98E+1	5.47E+2	3.17E+2	4.80E+2	7.46E+1
	Rank	2	1	6	4	5	3
29	Mean	3.06E+6	2.71E+2	2.15E+7	1.34E+7	4.38E+6	9.39E+5
	Std	3.62E+6	5.04E+2	1.09E+7	7.63E+6	4.68E+6	4.99E+5
	Rank	3	1	6	5	4	2
30	Mean	5.89E+4	1.18E+7	4.79E+5	2.48E+5	8.20E+4	4.37E+4
	Std	5.40E+4	8.87E+4	3.77E+5	8.39E+4	6.61E+4	1.71E+4
	Rank	2	6	5	4	3	1
Average rank		3.13	3.63	5.23	3.97	2.57	2.30
Overall rank		3	4	6	5	2	1

Table 7 Comparison results of MsRwGWO and GWO versions for 30D CEC2014 problems

No		OBGWO	IGWO	EEGWO	MsRwGWO	No		OBGWO	IGWO	EEGWO	MsRwGWO
1	Mean	3.19E+08	2.71E+08	1.77E+09	5.53E+07	16	Mean	1.40E+01	1.21E+01	1.40E+01	1.19E+01
	Rank	3	2	4	1		Rank	3	2	3	1
2	Mean	1.42E+10	1.93E+10	8.18E+10	2.97E+09	17	Mean	1.72E+07	7.30E+06	2.76E+08	1.46E+06
	Rank	2	3	4	1		Rank	3	2	4	1
3	Mean	8.81E+04	4.04E+04	1.08E+05	9.30E+03	18	Mean	2.39E+08	3.72E+07	7.62E+09	2.44E+07
	Rank	3	2	4	1		Rank	3	2	4	1
4	Mean	8.09E+02	1.25E+03	1.65E+04	2.56E+02	19	Mean	1.39E+02	1.39E+02	5.49E+02	2.79E+01
	Rank	2	3	4	1		Rank	2	2	3	1
5	Mean	2.12E+01	2.09E+01	2.12E+01	2.10E+01	20	Mean	2.61E+05	4.09E+04	4.74E+06	1.76E+03
	Rank	3	1	3	2		Rank	3	2	4	1
6	Mean	3.30E+01	1.99E+01	4.51E+01	2.03E+01	21	Mean	1.21E+07	2.90E+06	1.44E+08	4.00E+05
	Rank	3	1	4	2		Rank	3	2	4	1
7	Mean	9.51E+01	1.87E+02	8.27E+02	2.57E+01	22	Mean	1.20E+03	5.61E+02	5.72E+04	4.52E+02
	Rank	2	3	4	1		Rank	3	2	4	1
8	Mean	4.32E+02	1.31E+02	3.93E+02	1.68E+02	23	Mean	4.09E+02	3.73E+02	2.00E+02	3.26E+02
	Rank	4	1	3	2		Rank	4	3	1	2
9	Mean	3.12E+02	1.48E+02	3.99E+02	1.97E+02	24	Mean	2.02E+02	2.02E+02	2.01E+02	2.00E+02
	Rank	3	1	4	2		Rank	3	3	2	1
10	Mean	7.29E+03	3.01E+03	8.48E+03	4.74E+03	25	Mean	2.00E+02	2.05E+02	2.00E+02	2.10E+02
	Rank	3	1	4	2		Rank	1	2	1	3
11	Mean	7.08E+03	3.42E+03	9.07E+03	6.48E+03	26	Mean	1.48E+02	1.01E+02	1.96E+02	1.10E+02
	Rank	3	1	4	2		Rank	3	1	4	2
12	Mean	2.75E+00	2.35E+00	5.06E+00	2.46E+00	27	Mean	1.04E+03	8.04E+02	2.00E+02	7.23E+02
	Rank	3	1	4	2		Rank	4	3	1	2
13	Mean	2.90E+00	3.22E+00	8.76E+00	6.53E−01	28	Mean	2.50E+03	1.38E+03	2.00E+03	1.24E+03
	Rank	2	3	4	1		Rank	4	2	3	1
14	Mean	3.38E+01	5.11E+01	3.06E+02	4.52E+00	29	Mean	1.38E + 08	6.96E+06	2.00E+05	9.39E+05
	Rank	2	3	4	1		Rank	4	3	1	2
15	Mean	1.67E+03	9.51E+03	4.61E+05	3.68E+01	30	Mean	3.79E+06	1.62E+05	4.14E+05	4.37E+04
	Rank	2	3	4	1		Rank	4	2	3	1
						Average rank		2.9	2.1	3.3	1.4
						Overall rank		3	2	4	1

algorithm in 50 runs and error performance analyses are shown in Fig. 17a, b for short-term wind speed forecasting. As can be seen from these graphs, it is seen that MLP model, which has the best parameter optimized by MsRwGWO algorithm, gives successful performance in 1-h wind speed estimation in test and training stages. Also MsRwGWO-based MLP model performance is shown for training and test phases in Fig. 18 as scatter plots.

As shown in Fig. 17, prediction model performs poorly at overshoot points where wind speed changes suddenly. However, error performance metrics are needed to demonstrate the overall performance of the proposed MsRwGWO-MLP model against traditional GWO-MLP model. In practice, the forecasting capability of the proposed models can be evaluated by multiple statistical indices between the predicted and observed wind speed time series.

In this paper, the root-mean-squared error (RMSE), mean absolute percentage error (MAPE), mean-squared error (MSE), and mean absolute error (MAE) are utilized to evaluate the model performance. Generally, the smaller these performance metrics are, the better the model performs. These three performance metric indexes are calculated as follows [85, 86]:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{19}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{20}$$

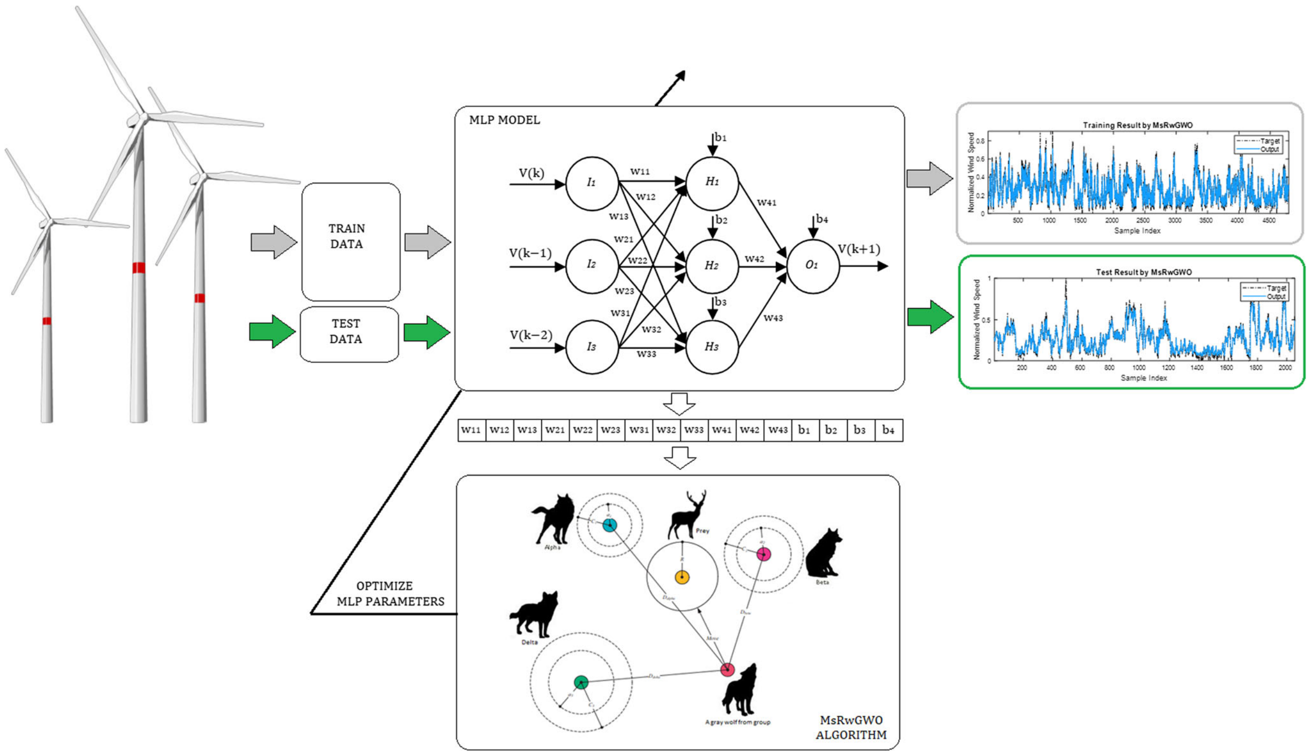


Fig. 16 MsRwGWO-based MLP hybrid model for wind speed forecasting

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{21}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{22}$$

where $\widehat{y}_i$ and y_i represent predictive and observed values of the wind speed, and N is the total number of data used for performance evaluation and comparison.

For comparison, forecasting training and test results of the traditional GWO-MLP model and the MsRwGWO-MLP model is shown in Fig. 19, respectively. In order to present the model performance more clearly, convergence curves of the both algorithms during the training MLP model are shown in Fig. 20. Here, the RMSE values of MLP models with the best parameter value obtained from 50 runs are presented depending on the iterations. From the zoom window shown at the end of iteration, it can be said that the proposed MsRwGWO algorithm converges better than the original GWO algorithm.

Although the both of the MLP models show a competitive result from figures, we presented Tables 8 and 9 summarizing the training and test results of MLP models with the 50 independent runs for both algorithms to better evaluate the performance of the proposed MsRwGWO algorithm. These tables have statistical results of MSE, RMSE, MAE, and MAPE metrics found by MsRwGWO-based MLP model and GWO-based MLP model for wind speed estimation. The best in these metrics has been emphasized in boldface. As can be seen from the training results in Table 8, MsRwGWO-MLP model gives better results than GWO-MLP model for all error metrics. Also, MsRwGWO-MLP model is the best in terms of all statistical metrics except of the standard deviation. From the test results given in Table 9, it is observed that for the MsRwGWO-based MLP model, the MSE, RMSE, MAE, and MAPE for mean performance metrics are 3.95E–3, 6.28E−2, 4.53E−2, and 20.8%, respectively. According to the best, mean, and median statistical metrics, the MLP model based on MsRwGWO algorithm has the better results than the other MLP model for the test part of the wind speed dataset. From the table, it can be confirmed that the proposed model achieves lower error values compared to GWO-MLP presented in these analysis results. The fact that the standard deviation of GWO is lower than that of MsRwGWO shows that the positions of the gray wolves are closer to each other in the GWO solution in the search space.

Finally, the comparison results of MsRwGWO algorithm and standard neural network training methods are summarized in Table 10. The classic methods used in this table are Gradient Descent with Momentum (GDM), Gradient Descent with momentum and adaptive learning rate (GDX), Conjugate Gradient with Polak-Ribiére updates

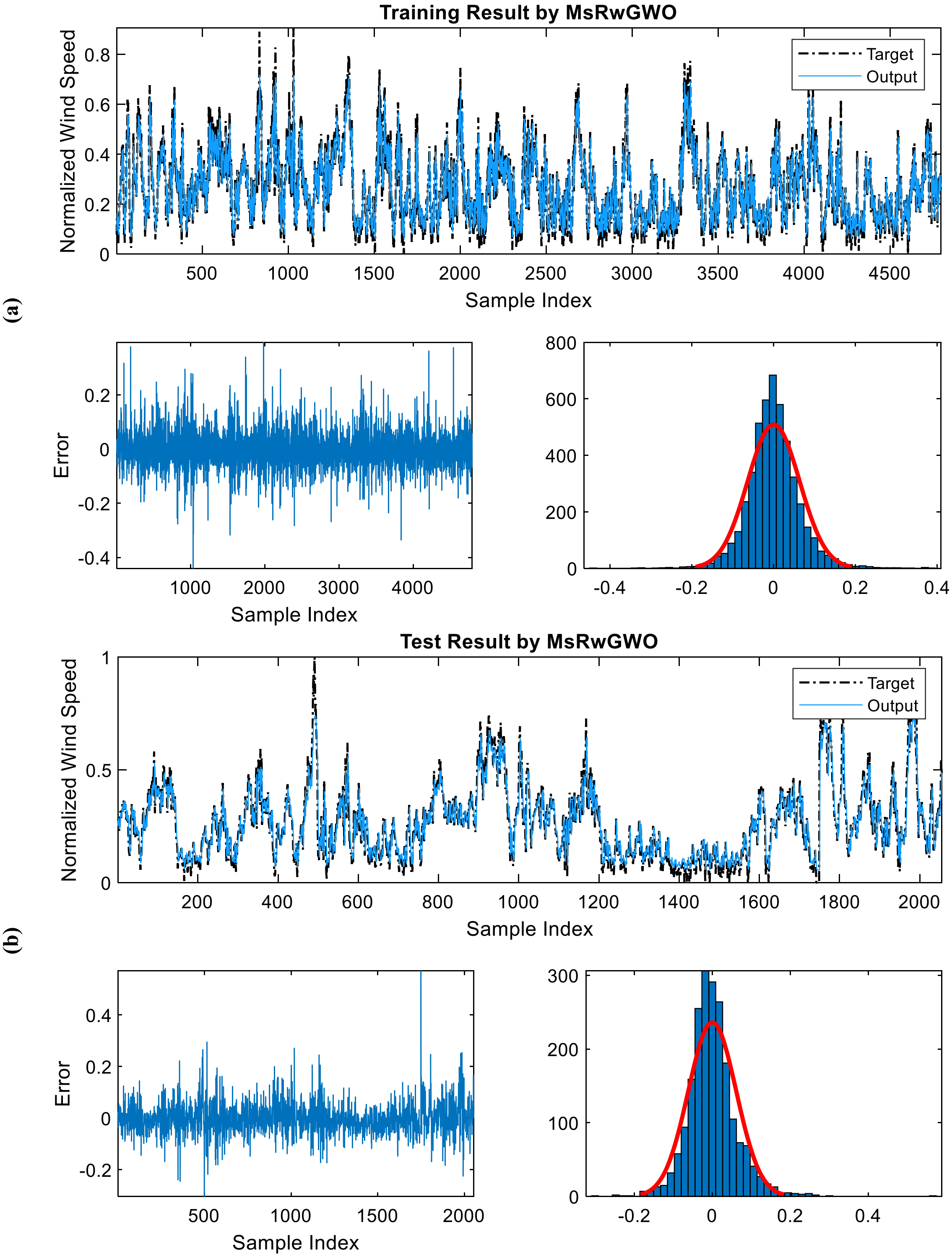


Fig. 17 MLP model with the best parameter obtained by MsRwGWO in 50 runs **a** training and **b** test results with statistical errors

(CGP), Conjugate Gradient with Powell-Beale restarts (CGB), One-Step Secant (OSS), BFGS quasi-Newton (BFG), Gradient Descent (GD), Gradient descent with adaptive learning rate (GDA), and Conjugate Gradient with Fletcher-Reeves updates (CGF) back propagation methods. This table has the results of MSE, RMSE, MAE, MAPE metrics, and training times. The best error metrics and algorithm duration are shown in bold in the table.

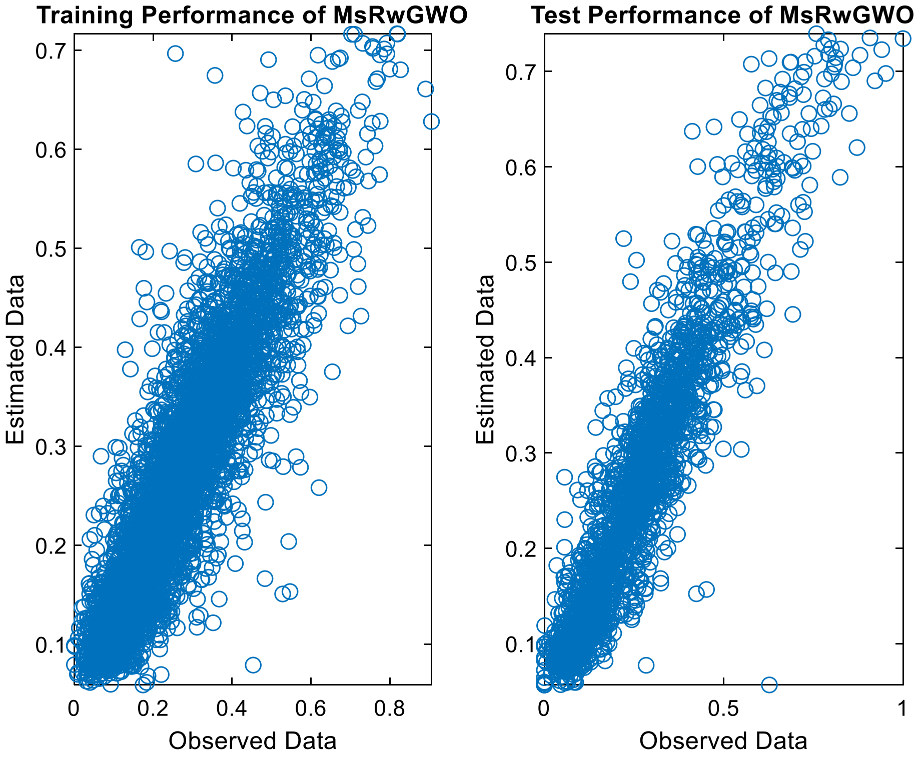


Fig. 18 Training and test performances of MsRwGWO-based MLP model

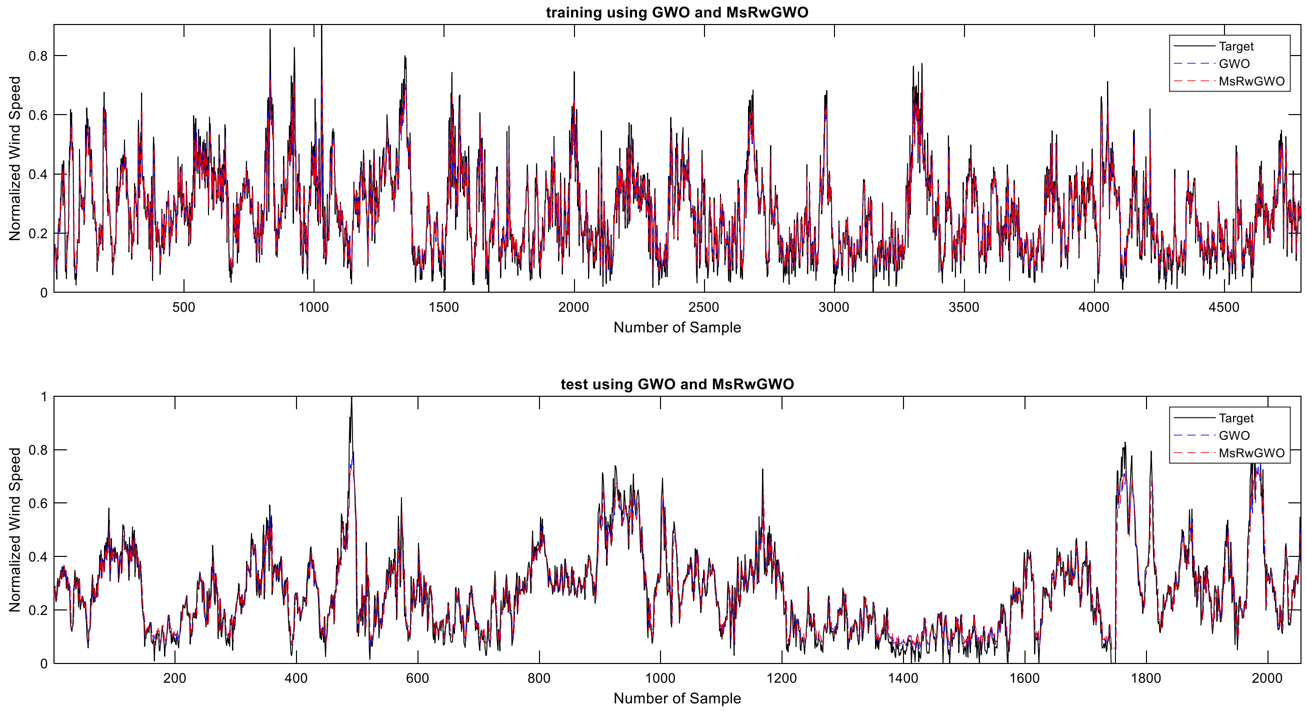


Fig. 19 Comparative training and test results of GWO and MsRwGWO-based MLP models

The training times of all algorithms were obtained for only one run. Note that the training time of GWO is higher than with traditional methods and MsRwGWO. As can be seen from Table 10, the proposed MsRwGWO has the best performance in terms of all metrics, but, as expected, its training time is higher than the other training methods. However, since the training of the model is generally done once, this long training time is not as important as expected in real-world problems.

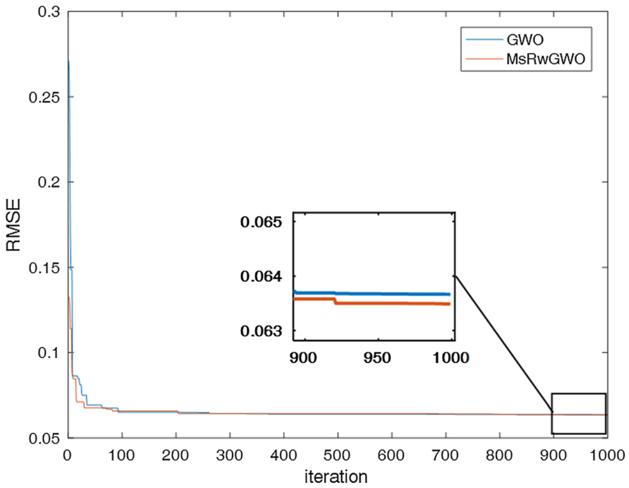


Fig. 20 Convergence curves of the MsRwGWO and GWO in the training of the MLP model

5 Conclusion

In this paper, a new GWO variant is proposed, named MsRwGWO, which presented a novel approach based on multi-strategy random weighted of GWO. The performance of the MsRwGWO is extensively analyzed based on three factors: (1) using convergence, search history, trajectory, and average distance analyses, (2) using CEC 2014 benchmarks with 10, 30, and 50 dimensions and some of the popular meta-heuristic algorithms such as MFO, PSO, DA, SCA, and WOA for 30D CEC 2014 test problems, (3) using the real-world problem like wind speed forecasting. As a result of the convergence analysis, it is seen that MsRwGWO has a faster convergence than the GWO algorithm in solving the problem. The ability of the proposed MsRwGWO to find a solution closer to the global optimum is seen. The results of search history analysis show that the distribution of gray wolves around the global optima, which are updated by the MsRwGWO algorithm, is higher than the distribution of gray wolves updated by the GWO algorithm in the search space during the exploration and exploitation phases. The gray wolves found by the GWO algorithm are stuck on the boundary values of the search space, especially on the surfaces of the benchmark problems except for the FN6 test problem in the search history analyses process. According to the trajectory analysis results of MsRwGWO algorithm, the alpha wolf's position is faster updated in the exploration stage and it gets closer to global optima in the exploitation stage. The proposed algorithm successfully avoids the local optimum points of the problem in the parts that show an increase in the average distance curve of the MsRwGWO algorithm during the exploration phase. Tests on CEC2014 show the MsRwGWO is a promising algorithm. At the same time, MsRwGWO algorithm is observed to perform better than MFO, PSO, DA, SCA, and WOA. In addition, the hybrid approach MsRwGWO-MLP model gives better results than GWO-MLP model for wind speed forecasting. The analyses results demonstrate that the proposed MsRwGWO-MLP hybrid model is a promising wind power forecasting

Table 8 Training results of GWO-MLP and MsRwGWO-MLP with 50 runs

	Metric	Best	Worst	Median	Mean	Std
GWO	MSE	4.05E−3	4.25E−3	4.12E−3	4.12E−3	**3.97E−5**
	RMSE	6.37E−2	6.52E−2	6.42E−2	6.41E−2	**3.08E−4**
	MAE	**4.60E−2**	4.80E−2	4.67E−2	4.66E−2	3.78E−4
	MAPE	2.00E+1	2.03E+1	2.01E+1	**2.01E+1**	**7.40E−2**
MsRwGWO	MSE	**4.03E−3**	**4.21E−3**	**4.10E−3**	**4.10E−3**	4.35E−5
	RMSE	**6.35E−2**	**6.49E−2**	**6.41E−2**	**6.40E−2**	3.39E−4
	MAE	**4.60E−2**	**4.72E−2**	**4.65E−2**	**4.65E−2**	**3.66E−4**
	MAPE	**1.99E+1**	**2.02E+1**	**2.00E+1**	**2.01E+1**	7.63E−2

Table 9 Test results of GWO-MLP and MsRwGWO-MLP with 50 runs

	Metric	Best	Worst	Median	Mean	Std
GWO	MSE	3.75E−3	**4.25E−3**	4.03E−3	3.99E−3	**1.31E−4**
	RMSE	6.12E−2	**6.52E−2**	6.35E−2	6.31E−2	**1.04E−3**
	MAE	4.42E−2	4.76E−2	4.59E−2	4.56E−2	**8.43E−4**
	MAPE	**2.06E+1**	2.14E+1	2.09E+1	2.09E+1	**1.44E−1**
MsRwGWO	MSE	**3.73E−3**	4.26E−3	**3.97E−3**	**3.95E−3**	1.50E−4
	RMSE	**6.11E−2**	6.53E−2	**6.30E−2**	**6.28E−2**	1.19E−3
	MAE	**4.41E−2**	**4.71E−2**	**4.55E−2**	**4.53E−2**	9.12E−4
	MAPE	**2.06E+1**	**2.12E+1**	**2.08E+1**	**2.08E+1**	1.47E−1

Table 10 Training performance results of MsRwGWO, GWO, and classic methods

Training method	MSE	RMSE	MAE	MAPE	Time (s)
MsRwGWO	**4.03E−03**	**6.35E−02**	**4.60E−02**	**1.99E+01**	92.929
GWO	4.05E−03	6.37E−02	**4.60E−02**	2.00E+01	120.253
GDX	4.17E−03	6.46E−02	4.71E−02	2.01E+01	7.559
CGP	5.47E−03	7.39E−02	5.43E−02	2.35E+01	0.938
CGB	6.72E−03	8.20E−02	6.39E−02	2.44E+01	0.824
OSS	4.13E−03	6.43E−02	4.68E−02	2.02E+01	5.464
GDM	5.33E−02	2.31E−01	1.91E−01	1.38E+02	6.725
BFG	4.06E−03	6.37E−02	4.61E−02	2.00E+01	5.637
GD	8.18E−03	9.05E−02	7.26E−02	2.91E+01	8.674
GDA	4.42E−03	6.65E−02	4.84E−02	2.04E+01	7.498
CGF	1.70E−02	1.30E−01	1.03E−01	3.71E+01	**0.511**

method, and it has higher forecasting accuracy and stronger stability. It is planned to develop hybrid models with decomposition methods in the future by incorporating correlated features in to input values.

Declarations

Conflict of interest The authors report no conflicts of interest. The authors alone are responsible for the content and writing of this article.

References

1. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley
2. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE international conference on neural networks, 1995. Proceedings, pp 1942–1948
3. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No.98TH8360), pp 69–73
4. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput J 8:687–697. https://doi.org/10.1016/j.asoc.2007.05.007
5. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. Artif Intell Rev 42:21–57. https://doi.org/10.1007/s10462-012-9328-0
6. Nanda SJ, Panda G (2014) A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol Comput 16:1–18. https://doi.org/10.1016/j.swevo.2013.11.003
7. Tayarani MHN, Yao X, Xu H (2015) Meta-heuristic algorithms in car engine design: a literature survey. IEEE Trans Evol Comput 19:609–629. https://doi.org/10.1109/TEVC.2014.2355174
8. Beheshti Z, Shamsuddin SMH (2013) A review of population-based meta-heuristic algorithm. Int J Adv Soft Comput Appl 5(1):1–35
9. Holland JH (1975) Adaptation in natural and artificial systems. Ann Arbor MI Univ Michigan Press Ann Arbor, p 183. https://doi.org/10.1137/1018105
10. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359. https://doi.org/10.1023/A:1008202821328
11. Price KV, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Springer Science & Business Media
12. Yüzgeç U, Becerikli Y, Türker M (2006) Nonlinear predictive control of a drying process using genetic algorithms. ISA Trans 45:589–602. https://doi.org/10.1016/S0019-0578(07)60234-1
13. Eusuff M, Lansey K, Pasha F (2006) Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Eng Optim 38:129–154. https://doi.org/10.1080/03052150500384759
14. Bhattacharjee KK, Sarmah SP (2014) Shuffled frog leaping algorithm and its application to 0/1 knapsack problem. Appl Soft Comput J 19:252–263. https://doi.org/10.1016/j.asoc.2014.02.010
15. Yang XS (2009) Harmony search as a metaheuristic algorithm. Stud Comput Intell 191:1–14
16. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. SIMULATION 76:60–68. https://doi.org/10.1177/003754970107600201
17. Gendreau M, Iori M, Laporte G, Martello S (2006) A Tabu search algorithm for a routing and container loading problem. Transp Sci 40:342–350. https://doi.org/10.1287/trsc.1050.0145
18. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(80):671–680. https://doi.org/10.1126/science.220.4598.671
19. Vinet L, Zhedanov A (2011) A 'missing' family of classical orthogonal polynomials. J Phys A Math Theor 44:085201. https://doi.org/10.1088/1751-8113/44/8/085201
20. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1:28–39. https://doi.org/10.1109/MCI.2006.329691
21. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B Cybern 26:29–41. https://doi.org/10.1109/3477.484436
22. Li X, Shao Z, Qian J (2002) An optimizing method based on autonomous animats: fish-swarm algorithm. Syst Eng Theory Pract 22:32–38
23. Timmis J, Hone A, Stibor T, Clark E (2008) Theoretical advances in artificial immune systems. Theor Comput Sci 403:11–32. https://doi.org/10.1016/j.tcs.2008.02.011
24. Timmis J, Andrews P, Hart E (2010) On artificial immune systems and swarm intelligence. Swarm Intell 4:247–273. https://doi.org/10.1007/s11721-010-0045-5

25. Das S, Biswas A, Dasgupta S, Abraham A (2009) Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. Found Comput Intell 3(3):23–55. https://doi.org/10.1007/978-3-642-01085-9_2
26. Li MS, Ji TY, Tang WJ et al (2010) Bacterial foraging algorithm with varying population. BioSystems 100:185–197. https://doi.org/10.1016/j.biosystems.2010.03.003
27. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci (Ny) 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
28. Sabri NM, Puteh M, Mahmood MR (2013) A review of gravitational search algorithm. Int J Adv Soft Comput Appl 5(3):1–39
29. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12:702–713. https://doi.org/10.1109/TEVC.2008.919004
30. Lim WL, Wibowo A, Desa MI, Haron H (2016) A biogeography-based optimization algorithm hybridized with tabu search for the quadratic assignment problem. Comput Intell Neurosci. https://doi.org/10.1155/2016/5803893
31. Sang HY, Duan PY, Li JQ (2018) An effective invasive weed optimization algorithm for scheduling semiconductor final testing problem. Swarm Evol Comput 38:42–53. https://doi.org/10.1016/j.swevo.2017.05.007
32. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022
33. Yang XS, Deb S (2014) Cuckoo search: recent advances and applications. Neural Comput Appl 24:169–174
34. Yang X-S (2009) Cuckoo Search via Lévy flights. In: 2009 world congress on nature and biologically inspired computing (NaBIC), pp 210–214
35. Heidari AA, Mirjalili S, Faris H et al (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872. https://doi.org/10.1016/j.future.2019.02.028
36. Bairathi D, Gopalani D (2020) A novel swarm intelligence based optimization method: Harris hawk optimization. In: Advances in intelligent systems and computing, pp 832–842
37. Coello Coello CA, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 36:219–236. https://doi.org/10.1080/03052150410001647966
38. Soza C, Becerra RL, Riff MC, Coello Coello CA (2011) Solving timetabling problems using a cultural algorithm. Appl Soft Comput 11:337–344. https://doi.org/10.1016/j.asoc.2009.11.024
39. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
40. Kılıç H, Yüzgeç U (2019) Improved antlion optimization algorithm via tournament selection and its application to parallel machine scheduling. Comput Ind Eng 132:166–186. https://doi.org/10.1016/j.cie.2019.04.029
41. Iscan H, Gunduz M (2014) Parameter analysis on fruit fly optimization algorithm. J Comput Commun 2:137–141. https://doi.org/10.1109/SITIS.2015.55
42. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
43. Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. Expert Syst Appl 166:113917. https://doi.org/10.1016/j.eswa.2020.113917
44. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004
45. Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE congress on evolutionary computation, CEC 2007, pp 4661–4667
46. Hosseini S, Al Khaled A (2014) A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research. Appl Soft Comput J 24:1078–1094
47. Yang XS (2007) Firefly algorithm. Nature-inspired metaheuristic algorithms, pp 79–90
48. Yang XS (2010) Firefly algorithm, Lévy flights and global optimization. In: Research and development in intelligent systems XXVI: incorporating applications and innovations in intelligent systems XVII, pp 1–10
49. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
50. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27:1053–1073. https://doi.org/10.1007/s00521-015-1920-1
51. Mafarja M, Aljarah I, Heidari AA et al (2018) Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowl Based Syst 161:185–204. https://doi.org/10.1016/j.knosys.2018.08.003
52. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
53. Barman M, Dev Choudhury NB (2020) A similarity based hybrid GWO-SVM method of power system load forecasting for regional special event days in anomalous load situations in Assam, India. Sustain Cities Soc. https://doi.org/10.1016/j.scs.2020.102311
54. Dogan L, Yüzgeç U (2018) Robot path planning using gray wolf optimizer. In: International conference on advanced technologies, computer engineering and science (ICATCES'18), pp 69–74
55. Karakas M, Yüzgeç U (2019) Opposition based gray wolf algorithm for feature selection in classification problems. In: 3rd International symposium on multidisciplinary studies and innovative technologies, ISMSIT 2019—proceedings. Institute of Electrical and Electronics Engineers Inc
56. Madadi A, Motlagh MM (2014) Optimal control of DC motor using grey wolf optimizer algorithm. Tech J Eng Appl 4:373–379
57. Medjahed SA, Ait Saadi T, Benyettou A, Ouali M (2016) Gray wolf optimizer for hyperspectral band selection. Appl Soft Comput J 40:178–186. https://doi.org/10.1016/j.asoc.2015.09.045
58. Li L, Sun L, Guo J et al (2017) Modified discrete grey wolf optimizer algorithm for multilevel image thresholding. Comput Intell Neurosci. https://doi.org/10.1155/2017/3295769
59. Ge L, Xian Y, Yan J et al (2020) A hybrid model for short-term PV output forecasting based on PCA-GWO-GRNN. J Mod Power Syst Clean Energy 8:1268–1275. https://doi.org/10.35833/MPCE.2020.000004
60. Liu H, Wu H, Li Y (2018) Smart wind speed forecasting using EWT decomposition, GWO evolutionary optimization, RELM learning and IEWT reconstruction. Energy Convers Manag 161:266–283. https://doi.org/10.1016/j.enconman.2018.02.006
61. Niu T, Wang J, Zhang K, Du P (2018) Multi-step-ahead wind speed forecasting based on optimal feature selection and a modified bat algorithm with the cognition strategy. Renew Energy 118:213–229. https://doi.org/10.1016/j.renene.2017.10.075
62. Xiao L, Wang J, Dong Y, Wu J (2015) Combined forecasting models for wind energy forecasting: a case study in China. Renew Sustain Energy Rev 44:271–288. https://doi.org/10.1016/j.rser.2014.12.012
63. Zhang W, Qu Z, Zhang K et al (2017) A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting. Energy Convers Manag 136:439–451. https://doi.org/10.1016/j.enconman.2017.01.022

64. Wang J, Du P, Niu T, Yang W (2017) A novel hybrid system based on a new proposed algorithm—multi-objective whale optimization algorithm for wind speed forecasting. Appl Energy 208:344–360. https://doi.org/10.1016/j.apenergy.2017.10.031
65. Osório GJ, Matias JCO, Catalão JPS (2015) Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information. Renew Energy 75:301–307. https://doi.org/10.1016/j.renene.2014.09.058
66. Fei SW, He Y (2015) Wind speed prediction using the hybrid model of wavelet decomposition and artificial bee colony algorithm-based relevance vector machine. Int J Electr Power Energy Syst 73:625–631. https://doi.org/10.1016/j.ijepes.2015.04.019
67. Rahmani R, Yusof R, Seyedmahmoudian M, Mekhilef S (2013) Hybrid technique of ant colony and particle swarm optimization for short term wind energy forecasting. J Wind Eng Ind Aerodyn 123:163–170. https://doi.org/10.1016/j.jweia.2013.10.004
68. Altan A, Karasu S, Zio E (2021) A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106996
69. Fu W, Wang K, Tan J, Zhang K (2020) A composite framework coupling multiple feature selection, compound prediction models and novel hybrid swarm optimizer-based synchronization optimization strategy for multi-step ahead short-term wind speed forecasting. Energy Convers Manag. https://doi.org/10.1016/j.enconman.2019.112461
70. Wang J, Wang S, Yang W (2019) A novel non-linear combination system for short-term wind speed forecast. Renew Energy 143:1172–1192. https://doi.org/10.1016/j.renene.2019.04.154
71. Wu C, Wang J, Chen X et al (2020) A novel hybrid system based on multi-objective optimization for wind speed forecasting. Renew Energy 146:149–165. https://doi.org/10.1016/j.renene.2019.04.157
72. Singh D, Dhillon JS (2019) Ameliorated grey wolf optimization for economic load dispatch problem. Energy 169:398–419. https://doi.org/10.1016/j.energy.2018.11.034
73. Pradhan M, Roy PK, Pal T (2016) Grey wolf optimization applied to economic load dispatch problems. Int J Electr Power Energy Syst 83:325–334. https://doi.org/10.1016/j.ijepes.2016.04.034
74. Jayabarathi T, Raghunathan T, Adarsh BR, Suganthan PN (2016) Economic dispatch using hybrid grey wolf optimizer. Energy 111:630–641. https://doi.org/10.1016/j.energy.2016.05.105
75. Pradhan M, Roy PK, Pal T (2018) Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system. Ain Shams Eng J 9:2015–2025. https://doi.org/10.1016/j.asej.2016.08.023
76. Gupta S, Deep K, Mirjalili S, Kim JH (2020) A modified sine cosine algorithm with novel transition parameter and mutation operator for global optimization. Expert Syst Appl 154:113395. https://doi.org/10.1016/j.eswa.2020.113395
77. Liang JJ, Qu BY, Suganthan PN (2014) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization
78. Long W, Liang X, Cai S et al (2017) A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. Neural Comput Appl 28:421–438. https://doi.org/10.1007/s00521-016-2357-x
79. Long W, Jiao J, Liang X, Tang M (2018) An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. Eng Appl Artif Intell 68:63–80. https://doi.org/10.1016/j.engappai.2017.10.024
80. Tascikaraoglu A, Uzunoglu M (2014) A review of combined approaches for prediction of short-term wind speed and power. Renew Sustain Energy Rev 34:243–254. https://doi.org/10.1016/j.rser.2014.03.033
81. Ma Z, Chen H, Wang J et al (2020) Application of hybrid model based on double decomposition, error correction and deep learning in short-term wind speed prediction. Energy Convers Manag. https://doi.org/10.1016/j.enconman.2019.112345
82. Lei M, Shiyan L, Chuanwen J et al (2009) A review on the forecasting of wind speed and generated power. Renew Sustain Energy Rev 13:915–920. https://doi.org/10.1016/j.rser.2008.02.002
83. Chang W-Y (2014) A literature review of wind forecasting methods. J Power Energy Eng 02:161–168. https://doi.org/10.4236/jpee.2014.24023
84. Liu H, Li Y, Duan Z, Chen C (2020) A review on multi-objective optimization framework in wind energy forecasting techniques and applications. Energy Convers Manag. https://doi.org/10.1016/j.enconman.2020.113324
85. Dokur E (2020) Swarm decomposition technique based hybrid model for very short-term solar PV power generation forecast. Elektron ir Elektrotechnika 26:79–83. https://doi.org/10.5755/j01.eie.26.3.25898
86. Dokur E, Kurban M, Ceyhan S (2016) Hybrid model for short term wind speed forecasting using empirical mode decomposition and artificial neural network. In: ELECO 2015—9th international conference electrical engineering, pp; 420–423. https://doi.org/10.1109/ELECO.2015.7394591

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