PARMAKSIZ, H., YUZGEC, U., DOKUR, E. and ERDOGAN, N. 2023. Mutation based improved dragonfly optimization algorithm for a neuro-fuzzy system in short term wind speed forecasting. *Knowledge-based systems* [online], 268, article 110472. Available from: https://doi.org/10.1016/j.knosys.2023.110472

Mutation based improved dragonfly optimization algorithm for a neuro-fuzzy system in short term wind speed forecasting.

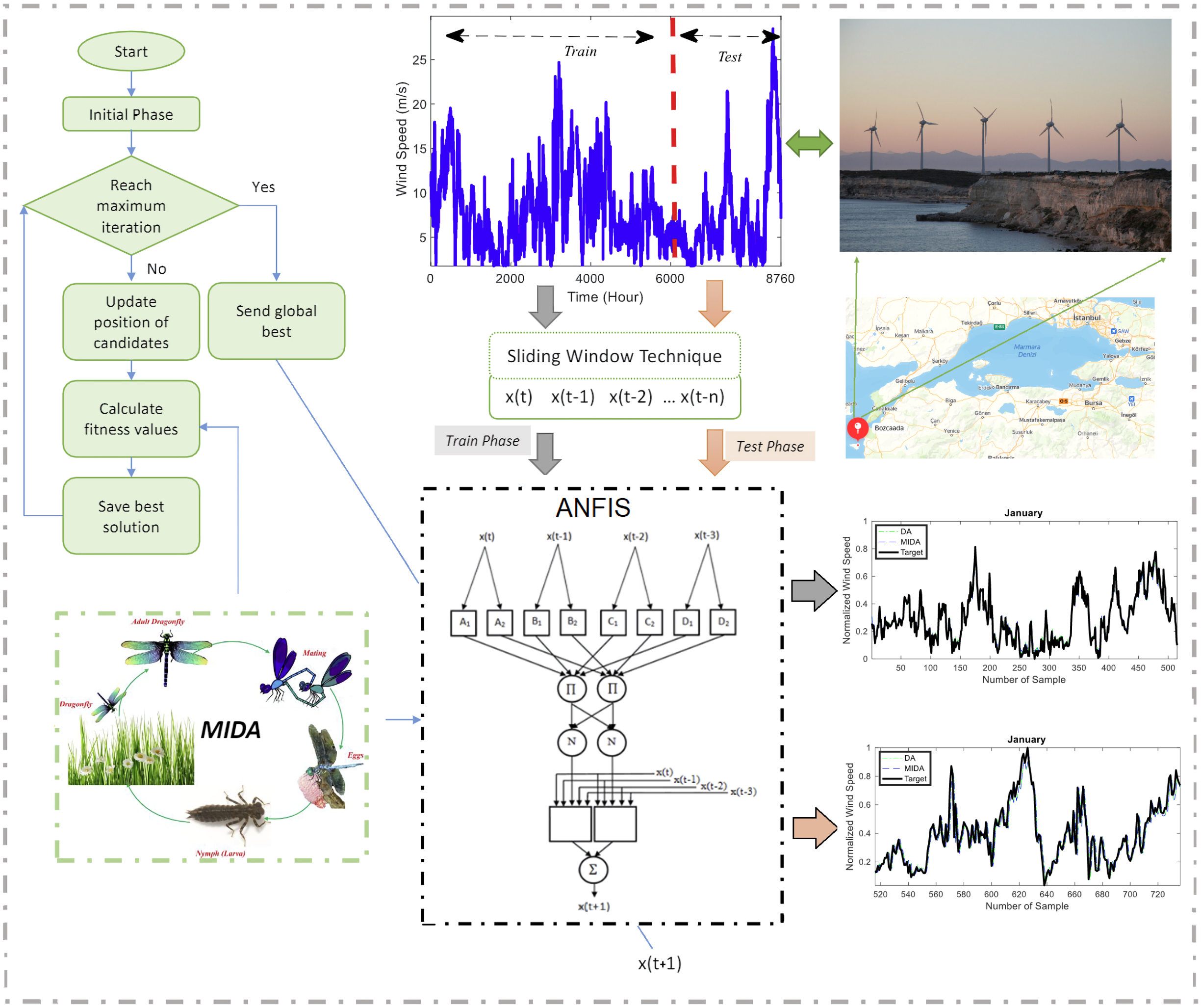
PARMAKSIZ, H., YUZGEC, U., DOKUR, E. and ERDOGAN, N.

2023

Graphical Abstract

Mutation based Improved Dragonfly Optimization Algorithm for a Neuro-Fuzzy System in Short Term Wind Speed Forecasting

Huseyin Parmaksiz,Ugur Yuzgec,Emrah Dokur,Nuh Erdogan



Highlights

Mutation based Improved Dragonfly Optimization Algorithm for a Neuro-Fuzzy System in Short Term Wind Speed Forecasting

Huseyin Parmaksiz,Ugur Yuzgec,Emrah Dokur,Nuh Erdogan

- A mutation-based dragonfly optimization algorithm (MIDA) is developed.
- The dragonfly algorithm is improved by incorporating three mechanisms.
- The proposed MIDA is evaluated using CEC2014 and CEC2020 benchmarks.
- MIDA was applied to optimize the parameters of a neuro-fuzzy model (ANFIS).
- A hybrid ANFIS-MIDA model is presented for short term wind speed forecasting.

Mutation based Improved Dragonfly Optimization Algorithm for a Neuro-Fuzzy System in Short Term Wind Speed Forecasting

Huseyin Parmaksiz[a], Ugur Yuzgec[a,*], Emrah Dokur[a,b] and Nuh Erdogan[c]

[a] Faculty of Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey
[b] Marine and Renewable Energy Centre,University of College Cork, Cork, Ireland
[c] School of Engineering, Robert Gordon University, Aberdeen AB10 7GJ, UK,

ARTICLE INFO

Keywords:
Dragonfly algorithm
Meta-heuristic
ANFIS
Forecasting
Wind speed

ABSTRACT

The Dragonfly algorithm (DA) is a heuristic optimization algorithm that is commonly used for complex optimization problems. Despite its widespread application, the abundance of social behaviors in its construct can lead to poor accuracy in solutions and an imbalance between exploration and exploitation phases. To overcome these issues, this paper proposes a mutation-based Dragonfly optimization algorithm (MIDA). In order to increase the solution accuracy of the original DA and reduce its handicaps, the proposed model includes three procedures, namely, mutation operation, boundary control, and greedy selection mechanisms. The mutation operator helps to find global optima by avoiding getting stuck at the local optimum point, while the boundary control and greedy selection mechanisms update the dragonflies at each iteration and thus use the better fitness value of the updated ones. The performance of the proposed MIDA is tested and shown to be superior to the original DA using several analyses such as convergence, search history, trajectory, average distance, computational complexity, diversity and balance. To validate the MIDA algorithm, it is tested on the 10, 30, 50 and 100 dimensions of the CEC2014 benchmark functions. Furthermore, a comparison against twelve state-of-the-art (SOTA) meta-heuristic optimization algorithms is performed. The performance of the proposed MIDA is also compared with the performances of some improved dragonfly algorithms taken from the literature. The statistical results obtained by the original DA and MIDA for ten minimization problems with 5, 10, 15 and 20 dimensions from the CEC2020 test suite are presented. Finally, the proposed algorithm is applied to optimize the ANFIS model parameters in order to be used for short-term wind forecasting as a real-world problem. The results obtained for the different dimensions of CEC2014 and CEC2020 test problems in the study show that the search performance of proposed MIDA is better than that of the original DA. MIDA outperformed the original DA by 91.33% for CEC2014 benchmarks and 94.25% for CEC2020 benchmarks. In addition, MIDA came first in comparison with twelve different SOTAs from the literature. In comparisons with different DA versions, MIDA took the second place in terms of statistical performance. Computational complexity was examined in the CEC2020 benchmarks and it was seen that MIDA has a more effective run time than DA. Finally, the short-term wind speed forecasting results of the ANFIS-DA hybrid model according to four different error metrics are more successful than those of the ANFIS-DA hybrid model.

1. Introduction

Meta-heuristic techniques are optimization algorithms that have been of interest to many researchers across a wide range of disciplines, including engineering, design optimization, business, and economics. They are now widely used to solve engineering problems for genuine reasons such as simplicity, flexibility, and robustness [1]. In terms of models, the meta-heuristic studies in the literature can be classified as non-nature-inspired meta-heuristics and nature-inspired meta-heuristics [2]. The first group algorithms include physics and/or chemical process-based, evolutionary-based, human-based, and swarm intelligence-based approaches.

Thanks to their flexibility, robustness, scalability, decentralised, self-organization, adaptation, and fast response, swarm intelligence (SI) based algorithms are becoming widely attractive in engineering fields in recent years [3], [4], [5], [6], [7]. These algorithms are essentially bio-inspired by the collective social behavior of swarms, systems, or communities, such as herds of animals, colonies of insects. The SI has been deemed an emerging field and an integral

*Corresponding author
ugur.yuzgec@bilecik.edu.tr (U. Yuzgec); edokur@ucc.ie (E. Dokur); n.erdogan@rgu.ac.uk (N. Erdogan)
ORCID(s):

part of artificial intelligence (AI). Some of the most well-known SI based algorithms are: Particle Swarm Optimizer (PSO) [8], Ant Colony Optimization algorithm (ACO) [9], Artificial Fish Swarm algorithm (AFS) [10], Bacterial Foraging Optimizer (BFO) [11], Artificial Bee Colony algorithm (ABC) [12], Firefly Algorithm (FA) [13], Ant Lion Optimizer (ALO) [14], Salp Swarm Algorithm (SWA) [15], Harris Hawks Optimizer (HHO) [16, 17], Tunicate Swarm Algorithm (TSA) [18, 19], Tree Seed Algorithm (TSA) [20, 21], Spotted Hyena Optimizer (SHO) [22, 23], African Vultures Optimization Algorithm (AVOA) [24], Artificial Gorilla Troops Optimizer (GTO) [25], Farmland Fertility Algorithm (FFA) [26, 27], and many others.

In 2016, Mirjalili proposed one of the most recent SI meta-heuristic techniques inspired by the dynamic and static swarming behaviors of dragonfly, called the Dragonfly Algorithm (DA) [28]. It is shown that the proposed DA algorithm in [28] can improve the initial random population, converge towards the global optimum, and provide very competitive results as compared to state-of-the-art algorithms. The DA has been applied to solve various optimization problems, including multi-objective problems. Thanks to its greater effectiveness and efficiency, the DA has been successfully applied to various applications, such as signal processing [29], robotics [30], power systems [31], biomedical [32], and many others. Compared to other optimization methods, the most significant advantages of DA are its simple, flexible and easily applicable structure and use of few control parameters [33]. However, the large number of social behavior models in the structure of DA can lead to decreased solution accuracy and an imbalance between the exploration and exploitation phases. Furthermore, the Levy flight model, which is used to increase the exploration capabilities of dragonflies in the search space, raises the risk that the updated positions of the dragonflies will go beyond the search space's boundaries. As a result, DA may fail to find global optima due to poor tuning of its control parameters [34].

There are four main techniques in the literature to improve the performance of DA. These are to improve the use of DA, improve local search capability, improve both the exploitation and exploration phases, and find better initial positions for dragonflies [35]. Yu et al.[36] presented an improved dragonfly algorithm based on two strategies, which is named QGDA. These strategies are Gaussian mutation (GM) and quantum rotation gate. The performance of QGDA was tested using CEC 2014 benchmark, feature selection, and several engineering design problems. In the study of Yuan et al. [37], the Gaussian mutation and the Gaussian bare-bone are integrated into the DA structure, which is called GGBDA. The benchmark result shows that the Gaussian mutation and Gaussian barebone structures effectively improved the performance of DA. Song et al. [38] proposed an improved dragonfly algorithm using elite opposition learning and exponential function, which is named EOEDA. The performance results of EOEDA are presented for 15 different benchmark problems. Yuan et al. [39] implemented an adaptive resistance and stamina strategy into the original dragonfly algorithm. The authors, calling their new algorithm ARSSDA, performed performance tests on 30 CEC2014 benchmarks and 3 well-known engineering problems. ARSSDA was inspired by the air resistance and physical endurance of insects in flight. In [40], a chaotic dragonfly optimizer (CDA) is presented for feature selection problems. The objective in the these problems is to minimize the size of the selected features, and so to maximize the performance of the classifiers. The results showed that the CDA outperformed the particle swarm optimization (PSO), cat swarm optimization (CSO), grey wolf optimization (GWO), and artificial bee colony (ABC) algorithms. Aci et al. [41] proposed a Brownian motion based DA for single and multi objective optimization problems. Brownian motion was used instead of Levy flight mechanism of the original DA. Bao et al. [42] improved a DA version using opposition-based learning, which is called OBLDA, to optimize the threshold value in multilevel threshold colour image segmentation. In the study of Sambandam et al. [43], a self adaptive scheme for tuning the parameters of DA was evaluated on the multilevel image segmentation problem. Salgotra et al. [44] considered different mutation operators (Lévy, trigonometric, Cauchy, diversity, Gaussian, neighbourhood-based, and clock mutations) to adapt it to the dragonfly algorithm. The experimental results (CEC2005, CEC2015, and CEC2011 benchmarks) show the mutation clock-based DA has the best performance among the other mutation-adapted DA versions.

In the literature, there are studies regarding hybrid structures that increase the optimization performance by using the dragonfly algorithm together with different meta-heuristic algorithms. Some of these are those: Abedi et al. [45] presented a hybrid algorithm consisting three different approaches. This hybrid structure is based on dragonfly algorithm, firefly algorithm, and opposition learning mechanism. In another study, Duan et al. [46] proposed a hybrid dragonfly algorithm with differential evolution algorithm for single objective optimization problems. In [47], a hybrid algorithm combining dragonfly and simulated annealing algorithms was developed for the flexible flow-shop scheduling problem. Ghanem et al. [48] presented a hybrid approach to optimize the weights of a multilayer perceptron model. This approach consists of dragonfly and artificial bee colony algorithms, which is called DA-ABC. Shilaja et al. [49] proposed a hybrid algorithm based on improved grey wolf optimizer and dragonfly algorithm for minimizing the energy cost, power loss and voltage deviation. The proposed hybrid algorithm was applied on IEEE 30 bus system and it was compared with some methods. In the study of Tawhid and Dsouza [50], a hybrid algorithm was introduced by combining dragonfly and particle swarm algorithms for the feature selection problems. The authors used UCI datasets to test the performance of hybrid DA-PSO. The results show that the proposed hybrid structure is more successful than other algorithms. Veeramsetty et al. [51] presented an approach based on hybridizing genetic and dragonfly algorithms. This hybrid algorithm was applied to the the 69 bus radial distribution system of a electric company.

In binary optimization problems, the search mechanisms of the dragonfly algorithm are adapted for such optimization problems. Some studies on the binary dragonfly algorithm are as follows: For solving the 0–1 knapsack problem, the binary dragonfly algorithm (BDA) was presented in [52]. The BDA was shown to be more accurate than the standard DA and other SI algorithms. Mafarja et al. [53] proposed a binary dragonfly algorithm (BDA) to minimize the number of features and increase the classification accuracy. Results with 18 UCI datasets show the superiority of BDA. In the study of Chen et al. [54], the single and multi binary dragonfly algorithms were presented for wavelength selection in the near-infrared spectroscopy analysis.

DA has recently been used in several machine learning (ML) models to specify the parameter's values of the predictors [55], [56], [57]. Among the ML models, the Adaptive Neuro-Fuzzy Inference System (ANFIS), a combination of artificial neural network (ANN) and fuzzy logic approaches regarded as one of the effective modeling methods [58]. Despite its widespread use, the ANFIS suffers from a significant computational complexity disadvantage. When the number of inputs increases, so does the number of tunable parameters. The main concern of researchers in designing the ANFIS model is how to update its parameters for effective forecasting. The meta-heuristic algorithms were used to tune the parameters of ANFIS in [59],[60],[61]. The hybrid meta-heuristic based ANFIS model has shown higher efficiency in several studies. PSO, GA and DE based ANFIS models were proposed to predict global solar radiation in [62]. Among the hybrid models considered, the hybrid ANFIS-PSO outperformed. Therefore, when the model is applied to real-world problems, determining ANFIS model parameters is one of the main issues encountered. Although the performance of the existing hybrid ANFIS models is promising, the forecasting capability still needs to be improved for practical implementations. As a result, an effective meta-heuristic method such as DA can be contributed to nonlinear dynamic systems.

One typical nonlinear dynamic system is a power system with a penetration of renewable energy sources such as wind. To enable adequate stability and reliability in such systems, fast and accurate wind forecasting tools are critical [63]. In recent years, meta-heuristic based AI models have grown in popularity in wind energy forecasting due to their high accuracy [64, 65, 66, 67]. The superiority of hybrid approaches using AI-based hybrid methods in terms of higher accuracy in wind speed estimation is demonstrated in [64]. Based on AI and decomposition methods with the GWO, a hybrid model was proposed in [68]. It is shown that the GWO based AI model outperforms its counterparts. Fuzzy system with Particle Swarm Optimization achieved higher accuracy and less uncertainty in the short-term wind speed predictions while having a low computation load in [69]. Zhang et al. in [70] proposed the multi-objective dragonfly algorithm (MODA) and decomposition with AI techniques for wind speed forecasting. The MODA based hybrid model demonstrated superior performance with higher accuracy over non meta-heuristic-based models. Based on location, the wind speed characteristics might differ, which cannot be captured by certain forecasting models. While meta-heuristic-based models have shown successful applicability to wind speed forecasting, some challenges such as premature convergence, being stuck at the local minima, overfitting, etc. are mostly encountered in the implementation. Therefore, novel meta-heuristic algorithms need to be explored to overcome these challenges.

Dragonfly optimization algorithm has the ability of superior exploration to explore the search region and to find promising areas. Nevertheless, the dragonfly algorithm has some drawbacks as given below:

- Nevertheless, the solutions found for the optimization problem may have low accuracy due to DA's low exploitation ability.

- At the same time, although DA has good global search capability for avoiding local optimal points in the search space, it is likely to still get stuck at local optima.

- The five different models of social behavior in DA code can lead to an imbalance between the exploration and exploitation stages, resulting in decreased solution accuracy.

In this study, a Mutation-based Improved Dragonfly optimization Algorithm (MIDA) is proposed to eliminate the handicaps of the dragonfly algorithm given above and to highlight the algorithm's ability, especially in the exploitation phase, and to increase the search capability in general. The proposed MIDA consists of three methods, namely, mutation operator, boundary control mechanism, and greedy selection mechanism. The mutation operator helps dragonflies find the global optimum by avoiding being stuck at the local optimum in the search space. The boundary control method has been added to provide a solution to the exceeding of the limit values of mechanisms such as Levy flight in the original DA, especially in the exploration phase. Finally, the greedy selection mechanism updates the dragonflies at the end of each iteration, thus using the better fitness value of the updated ones.

The MIDA was first evaluated using several analyses, such as convergence, complexity, trajectory, diversity, etc. The algorithm is then tested on some numerical optimization problems, also named the CEC 2014 test package in order to validate its performance. The original DA and proposed MIDA algorithms are used to solve benchmark test problems in 10, 30, 50, and 100 dimensions for 51 independent runs. Wilcoxon rank-sum tests are used to determine whether there is a statistically significant difference between the iterative optimization results of the original DA and MIDA. Furthermore, the MIDA is validated against state-of-the-art (SOTA) meta-heuristic algorithms, such as Particle Swarm Optimizer (PSO), Moth-Flame Optimizer (MFO), Sine-Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA),Chaotic Cuckoo Search optimizer (CCS), Coyote Optimization Algorithm (COA), Elephant Herding Optimizer (EHO), Ageist Spider Monkey Optimizer (ASMO), Variable Neighborhood Bat Algorithm (VNBA), Red Fox Optimizer (RFO), and Human Felicity Algorithm (HFA). The performance of the proposed MIDA is also compared with the performances of some improved dragonfly algorithms taken from the literature, such as Quantum-behaved and Gaussian mutational Dragonfly Algorithm (QGDA), Adaptive Resistance and Stamina Strategy-based Dragonfly Algorithm (ARSSDA), Elite Opposition learning and Exponential function adaptive steps based Dragonfly Algorithm (EOEDA), and Gaussian Mutational Bare-bone Dragonfly Algorithm (GGBDA). For CEC2020, one of the latest single-objective optimization tests, the original DA and MIDA were run 30 times for 10 minimization problems with 5, 10, 15, and 20 dimensions. In addition, the computational complexity of both algorithms is examined for the 5, 10, and 15 dimensional F1 function. All comparative results show that the proposed MIDA can be used as an alternative to other algorithms in real optimization problems in many different fields. Finally, the MIDA is adapted to optimize the parameters of the ANFIS model utilized in the wind speed forecasting problem. Model parameters were optimized using the proposed MIDA and original DA structures in the training of the ANFIS model, then short-term wind speed forecasting test results of ANFIS-DA and ANFIS-MIDA hybrid models were compared according to different error metrics (MSE, RMSE, MAE, and MAPE). All the error metric results show the superiority of the ANFIS-MIDA hybrid model over the ANFIS-DA model.

In the remainder of this paper, Section 2 presents the development of the proposed mutation based improved DA along with detailing the conventional DA and ANFIS models. The some analysis for MIDA, the benchmark results for CEC2014 and CEC2020, the performance of ANFIS-MIDA hybrid model for wind speed forecasting are given in Section 3. In Section 4, all the results obtained in this study are examined and the advantages and limitations of MIDA are discussed. Finally, Section 5 presents conclusions and future studies.

2. Material and methods

In this section, the model equations and Pseudo codes for the proposed method are developed. The dragonfly optimization algorithm is first introduced. The DA structure is then improved by incorporating a mutation operator. Finally, the model equations for ANFIS are presented.

2.1. Dragonfly optimization algorithm

The DA is a swarm-based meta-heuristic optimization algorithm that mimics the hunting and migration mechanisms of the dragonflies, proposed by Mirjalili in [28]. Dragonflies have two main purposes as swarm behaviour: hunting and migration. Dragonflies that collect to hunt represent a static swarm, and those that collect to migrate represent a dynamic swarm. Static and dynamic swarm behaviors correspond to exploration and exploitation phases in meta-heuristic optimization algorithms. A total of five basic swarm behavior models are used in the dragonfly algorithm. These are separation, alignment, cohesion, attraction to food, and distraction from enemies, respectively. Fig. 1 shows the swarm behaviour models of the dragonflies.

The mathematical models of the swarm behaviors can be given as follow:

$$S_i = -\sum_{j=1}^{N} \left(X - X_j \right) \tag{1}$$

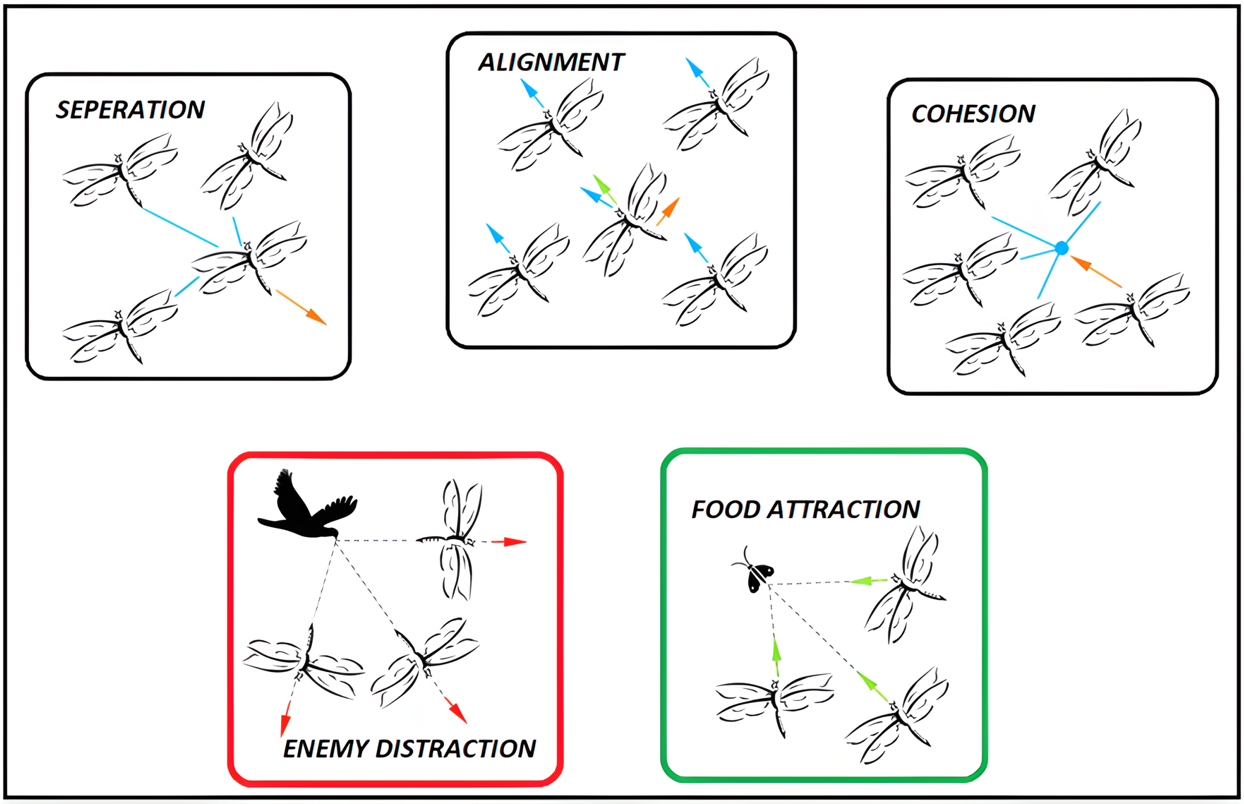


Figure 1: Swarm behavior models of dragonflies [28]

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{2}$$

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \tag{3}$$

$$F_i = X^+ - X \tag{4}$$

$$E_i = X^- + X \tag{5}$$

Where X_j represents the position of the j^{th} neighbouring dragonfly, X represents the current dragonfly's position, N is the number of neighbouring dragonflies, V_j represents the velocity of the neighbouring dragonfly, X^+ and X^- represent the position of the food source (best dragonfly) and the position of the enemy (worst dragonfly), respectively.

At each iteration, the positions of the dragonflies are updated by the step vector (ΔX_{t+1}) as given below:

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{6}$$

$$\Delta X_{t+1} = \left(sS_i + aA_i + cC_i + fF_i + eE_i\right) + w\Delta X_t \tag{7}$$

where s represents the weight value for separation, a is the weight value of the alignment process, c denotes the cohesion weight value, f is the food factor, e is the enemy factor, and w is the inertia weight. In the absence of neighboring solutions, a Lévy flight is used in the algorithm structure to improve the stochastic behavior and search discovery of the DA mechanism. The position of the dragonfly is updated by

$$X_{t+1} = X_t + L(D) \times X_t \tag{8}$$

where D is the dimension, L is the Lévy flight function. The Pseudo code of the DA is given in Algorithm 1.

Algorithm 1 Pseudo code of Dragonfly Algorithm

procedure DRAGONFLY($X_i, N_p, D, it_{max}, U_b, L_b$)
  $X_i \leftarrow rand_i(N_p, D)$ ▷ initialize dragonflies
  while $it \leq it_{max}$ **do**
    $Fit_i \leftarrow f(X_i)$ ▷ Calculate fitness values
    $X^+, X^- \leftarrow best(X_i), worst(X_i)$
    update weight parameters (s, a, c, f, e, w)
    calculate S_i, A_i, C_i, F_i, E_i
    if $N > 1$ **then** ▷ any neighbours
      $\Delta X_{t+1} \leftarrow S_i, A_i, C_i, F_i, E_i$
      $X_{t+1} \leftarrow X_t + \Delta X_{t+1}$
    else
      $X_{t+1} \leftarrow X_t$+Lévy$(D) \times X_t$
    end if
    check X_{t+1} based on boundary conditions
  end while
  return X^+
end procedure

2.2. Proposed mutation based improved Dragonfly algorithm (MIDA)

In the proposed method, first, a mutation operator was added to the original DA structure to avoid getting stuck in the local optimum after updating the positions of the dragonflies in each iteration. The equation for the mutation operator is as follow:

$$X_{t+1} = X_t + \sigma \times \left(U_b - L_b\right) \times r_m \tag{9}$$

where σ denotes the scale factor, U_b, L_b denote upper and lower boundaries, and r_m is the random number with normal distribution. The Pseudo code regarding the mutation procedure is given in Algorithm 2.

Algorithm 2 Pseudo code of the mutation procedure

if $rand(0, 1) < \beta$ **then**
  $j \leftarrow$ randsample(D, n_μ) ▷ select index from D
  $X_{t+1}(i, j) \leftarrow X_t(i, j) + \sigma \times \left(U_b - L_b\right) \times r_m$
end if

In Algorithm 2, β denotes the mutation probability (typically a value less than 0.01 is chosen), D is the problem dimension, n_μ represents the mutation length, which is less than the problem dimension, and σ is the sigma coefficient (i.e., 0.1). After updating meta-heuristic search agents, their new positions may exceed search space boundaries. If the search agent's updated position exceeds the search space boundary, the search agent is moved above the boundary value. This is a fairly common method in meta-heuristic algorithms, particularly in the exploration phase, and as a result, many search agents become stuck at the search space's boundaries in some problems. To avoid this situation, the DA structure is given a new boundary control mechanism. In this boundary control mechanism, the positions of the dragonflies are controlled if they are outside the search space areas. While updated position of the dragonfly is outside of the limit values, the dragonfly is randomized to a position within a maximum of 25% of the boundary values of the search space. The proposed boundary control mechanism equations are as follow:

$$X_{t+1} = \begin{cases} U_b - 0.25 \times \left(U_b - L_b\right) \times r_b, X_{t+1} > U_b \\ L_b + 0.25 \times \left(U_b - L_b\right) \times r_b, X_{t+1} < L_b \end{cases} \tag{10}$$

where r_b denotes the random number between 0-1. The original DA structure does not include any selection mechanism. The updated dragonflies are used directly in the next generation. Finally, a greedy selection mechanism is added to the DA structure. Thus, if the fitness values of the updated dragonflies at the end of each iteration are better, their new positions are used, otherwise the search continues with their old positions. This selection mechanism is given in Algorithm 3.

Algorithm 3 Pseudo code of selection procedure

if $f(X_t) < f(X_{t+1})$ **then**
  $X_{t+1} \leftarrow X_t$
  $f(X_{t+1}) \leftarrow f(X_t)$
end if

There are three basic mechanisms (mutation operator, boundary control mechanism, and greedy selection mechanism) integrated in the structure of the proposed MIDA. The Pseudo code of MIDA is given in Algorithm 4. The flowchart of the MIDA is shown in Fig.2.

Algorithm 4 Pseudo code of the mutation based improved Dragonfly algorithm (MIDA)

```
procedure MIDA(X_i, N_p, D, it_max, U_b, L_b)
    X_i ← rand_i(N_p, D)
    while it ≤ it_max do
        Fit_i ← f(X_i)
        X^+, X^- ← best(X_i), worst(X_i)
        update weight parameters (s, a, c, f, e, w)
        calculate S_i, A_i, C_i, F_i, E_i
        if N > 1 then
            ΔX_{t+1} ← S_i, A_i, C_i, F_i, E_i
            X_{t+1} ← X_t + ΔX_{t+1}
        else
            X_{t+1} ← X_t+Lévy(D) × X_t
        end if
        if rand(0, 1) < β then                                  ▷ Mutation operator
            j ← randsample(D, n_μ)
            X_{t+1}(i, j) ← X_t(i, j) + σ × (U_b − L_b) × r_m
        end if
        if X_{t+1} > U_b then                                   ▷ Boundary control
            X_{t+1} = U_b − 0.25 × (U_b − L_b) × r_b
        else if X_{t+1} < L_b then
            X_{t+1} = L_b + 0.25 × (U_b − L_b) × r_b
        end if
        if f(X_t) < f(X_{t+1}) then                             ▷ Selection mechanism
            X_{t+1} ← X_t
            f(X_{t+1}) ← f(X_t)
        end if
    end while
    return X^+
end procedure
```

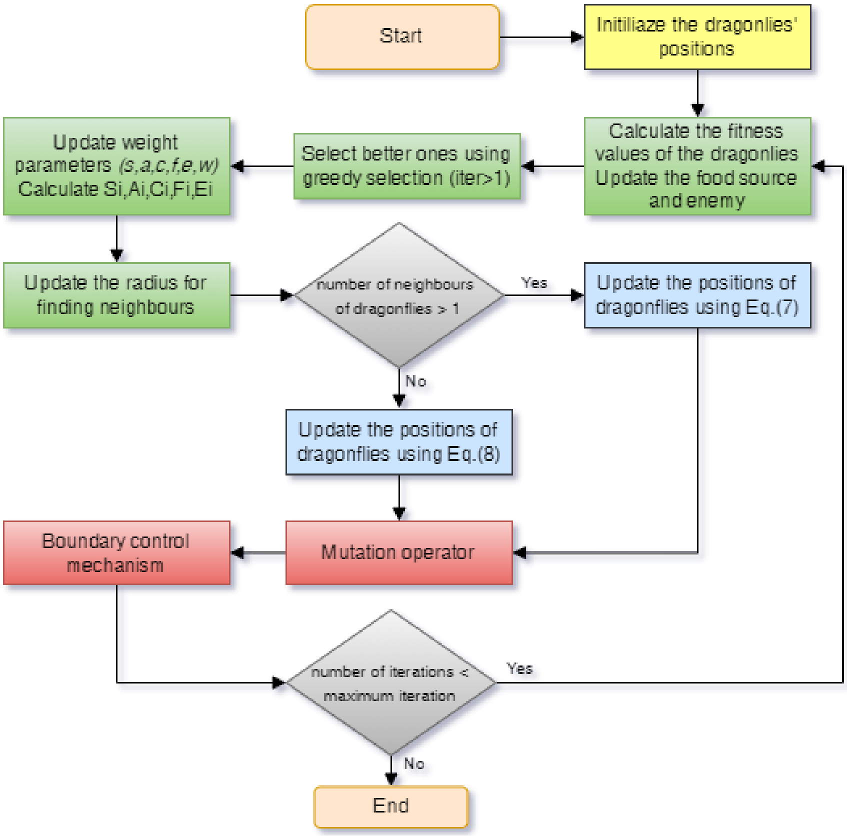


Figure 2: Flowchart of MIDA

2.3. Adaptive Neuro-Fuzzy Inference System (ANFIS) model

ANFIS was developed by Jang [71] in the early 1990s as a type of artificial neural network method based on the Takagi-Sugeno fuzzy inference system. This model is a hybrid neuro-fuzzy structure that is primarily used for modelling and controlling uncertain systems. The ANFIS model structure, in general, consists of five distinct layers, which can also be defined as a multilayer neural network. An ANFIS model structure with two inputs and one output is shown in Fig. 3. Based on the model architecture in Fig. 3, the general mathematical equations of the ANFIS model can be expressed as follows:

$$O_{1,i} = \mu_{A_i}(x) = \exp\left(-\frac{(x-a_i)^2}{2b_i^2}\right); i = 1,2 \tag{11}$$

$$O_{1,i} = \mu_{B_{i-2}}(y) = \exp\left(-\frac{(y-a_i)^2}{2b_i^2}\right); i = 3,4 \tag{12}$$

$$O_{2,i} = w_i = \mu_{A_i}(x)\,\mu_{B_i}(y), i = 1,2 \tag{13}$$

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \tag{14}$$

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i\left(p_i x + q_i y + r_i\right), i = 1,2 \tag{15}$$

$$O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, i = 1, 2 \tag{16}$$

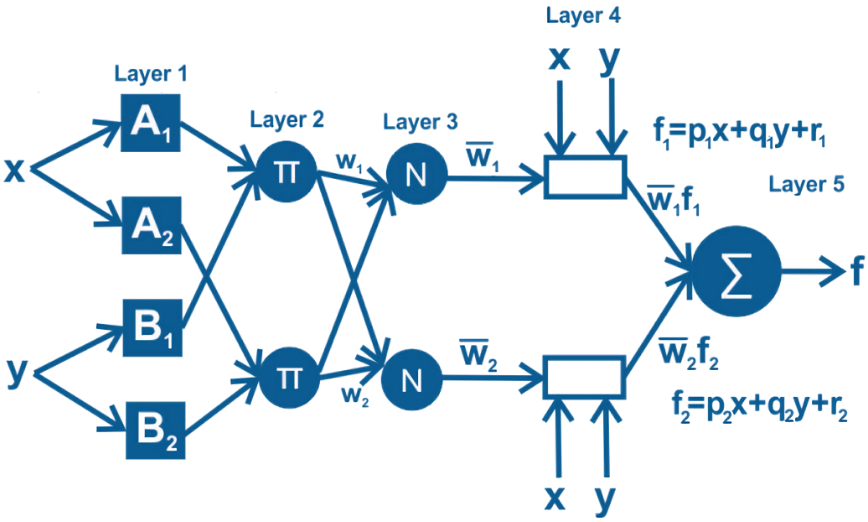


Figure 3: General ANFIS model architecture [72]

There are Gauss functions for each input in the first layer (Eq. 11-Eq. 12), and a_i and b_i in the Gauss function are called antecedent parameters.Eq. 13 shows the output of the second layer. Here, membership function values are multiplied using the algebraic product T-norm operator. Eq. 14 gives the output expression of the third layer and it is called normalized firing strengths. Here, the average of each node's firing strength is found. For each input, the node functions are calculated, and the output of this layer is found by multiplying it by the average rule firing strength, which is the output of the 3rd layer, as given in Eq. 15. The parameters in this layer (p_i, q_i, and r_i) are referred to as the consequent parameters. Eq. 17 gives the output calculation of the last layer. In this layer, a single output is obtained from the sum of the outputs from the previous layer.

The ANFIS model used for estimating the short-term wind speed in this study is a model with 4 inputs and 1 output. The inputs consist of the hourly wind speed instantaneous and its values one hour, two hours, and three hours ago. Again, two Gaussian functions are used for each input and output in the ANFIS model, and there are two polynomials in the defuzzification layer. This model with 4 inputs and 1 output used here is the one with the best result we obtained from different ANFIS models in our previous studies [60]. In the training of ANFIS model, a_i and b_i coefficients of Gaussian functions (antecedent parameters given in Eq. 11), and p_i, q_i, and r_i coefficients in defuzzification layer (consequent parameters given in Eq. 15) is being optimized. Since there are two Gaussian functions at each input in the model, a total of $4 \times 2 = 8$ Gaussian functions and $8 \times 2 = 16$ antecedent parameters from two parameters in each function, should be optimized. Likewise, since there are five consequent parameters for 4 inputs in 2 polynomials, the $2 \times 5 = 10$ consequent parameters are used in training. As a result, 26 ANFIS parameters, including 16 antecedent parameters and 10 consequent parameters, are optimized during the training phase. With the MIDA structure proposed in the ANFIS training, it is aimed to optimize these 26-dimensional parameters. Each of the dragonflies in MIDA is initialized with 26-dimensional random values for the 26 parameters of the ANFIS model. Each of the dragonflies in MIDA is initialized with 26-dimensional random values for the 26 parameters of the ANFIS model. The fitness value for each dragonfly is calculated with the RMSE value of the ANFIS model. With the help of updates and other mechanisms in MIDA, the 26-dimensional parameters of the best dragonfly in the population obtained after 1000 iterations are set to the ANFIS model. Then, the ANFIS model, which is set with the best parameters found by MIDA, is evaluated with the test data set.

Table 1
Summary of the CEC 2014 Test Functions [73]

No	Functions	Solution	No	Functions	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→ 3)	1700
FN3	Discus with rotated	300	FN18	Hybrid Func. 2 (N → 3)	1800
FN4	Rosenbrock with shifted & rotated	400	FN19	Hybrid Func. 3 (N → 4)	1900
FN5	Ackley with shifted & rotated	500	FN20	Hybrid Func. 4 (N → 4)	2000
FN6	Weierstrass with shifted & rotated	600	FN21	Hybrid Func. 5 (N →5)	2100
FN7	Griewank with shifted & rotated	700	FN22	Hybrid Func. 6 (N → 5)	2200
FN8	Rastrigin with shifted	800	FN23	Composition Func. 1 (N → 5)	2300
FN9	Rastrigin with shifted & rotated	900	FN24	Composition Func. 2 (N → 3)	2400
FN10	Schwefel with shifted	1000	FN25	Composition Func. 3 (N → 3)	2500
FN11	Schwefel with shifted & rotated	1100	FN26	Composition Func. 4 (N → 5)	2600
FN12	Katsuura with shifted & rotated	1200	FN27	Composition Func. 5 (N → 5)	2700
FN13	HappyCat with shifted & rotated	1300	FN28	Composition Func. 6 (N → 5)	2800
FN14	HGBat with shifted & rotated	1400	FN29	Composition Func. 7 (N → 3)	2900
FN15	F4 with expanded plus F7	1500	FN30	Composition Func. 8 (N → 3)	3000

3. Analysis results

In this section, the search capability of the proposed MIDA is compared to the original DA in terms of (i) convergence analysis, (ii) search history analysis, (iii) trajectory analysis, (iv) average distance analysis, (v) computational complexity analysis, (vi) balance analysis, and (vii) diversity analysis. The main purpose of these analyzes is to compare the search behaviours of the proposed MIDA and the original DA while seeking the global optimum in the search space. In the convergence analysis, the fitness value of the best dragonfly obtained during the optimization, the positions of all dragonflies in the search history analysis, the position of the elite dragonfly in the trajectory analysis, the average distance of the first dragonfly with the others in the average distance analysis are used. In computational complexity analysis, the amount of time the algorithm consumes when running is found with the O(.) notation. Balance analysis presents the search performance of algorithms in the exploration and exploitation stages. In diversity analysis, the average distance between dragonflies during the search for the global solution is used.

Second, the benchmark functions, known as IEEE Congress on Evolutionary Computation (CEC) 2014 in the literature, are used to test the performance of the proposed MIDA. The CEC2014 test suite consists of thirty benchmark problems with different features [73]. This test suite consists of different types of issues: simple multimodal, unimodal, composition, and hybrid structure. In Table 1, the benchmark functions and solutions in the test suite are summarized. The report by Liang et al.[73] contains detailed information about the CEC2014 test suite.

The original DA and MIDA solutions of the 10, 30, 50, and 100 dimensional benchmark problems were obtained with 51 runs, and these results were compared statistically. The CEC2014 results obtained by the proposed MIDA and the original DA were compared by using the Wilcoxon rank sum test, one of the non-parametric tests, for statistical significance. In addition, 30 dimensional CEC2014 test problems were compared with some up-to-date meta-heuristic algorithms from the literature to demonstrate the superiority of the proposed MIDA algorithm. The 30D CEC2014 results of the MIDA were also compared with the results of some improved versions of dragonfly algorithm from the literature. In addition to the CEC2014 benchmark results of the proposed MIDA and the original DA, the both algorithms are compared for 10 minimization problems in the CEC2020 benchmark package, one of the latest test problems in the field of numerical optimization. Here, the computational complexity of both algorithms were presented for F1 function from CEC2020 benchmarks. Finally, a real-world problem is discussed, and the proposed MIDA structure optimizes the model parameters in the training phase of an ANFIS model used in short-term wind speed forecasting. Wind speed estimation results from the training and testing phases are compared to the ANFIS-DA hybrid structure using various error metrics.

For the MIDA's analysis, four different benchmark functions from the CEC 2014 benchmark problems were selected. These functions are as follows: rotated high conditioned elliptic function (FN1) from Rastrigin's function

(FN9) and Griewank's plus Rosenbrock's function (FN15) and composition function 3 (N=3) (FN25). The 3D maps and contour plots of the benchmark functions selected for analysis of the MIDA are shown in Fig. 4.

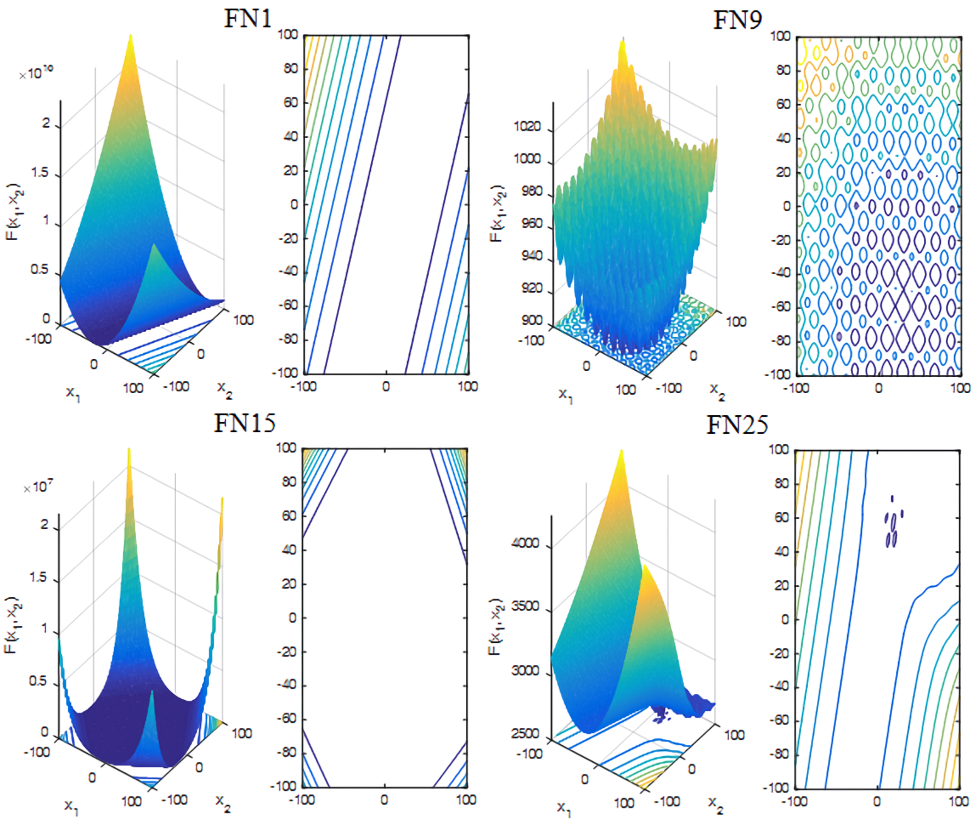


Figure 4: Unimodal, multimodal and composition benchmark functions used for performance analysis of the proposed MIDA

3.1. Convergence analysis

The convergence characteristic of an optimization algorithm gives us information about the rate of searching for the global optimum point of the algorithm. Thus, the performance of heuristic algorithms is highly dependent on the convergence behavior during solving optimization problems. In this context, four different comparison problems were taken from the CEC2014 test package, and convergence curves were obtained from the solutions of these problems with the proposed MIDA and original DA. In this analysis, the initial positions of the dragonflies were the same for both algorithms. Again, in this analysis, the problem size was chosen as two. Fig. 5 shows the convergence curves obtained by both algorithms on a logarithmic scale for the four benchmark functions.

These convergence curves display the residue values of the best dragonflies found by both algorithms (e.g., MIDA and DA) during the solution of optimization problems. The error value is found by taking the difference between the benchmark function and the best solution found by the meta-heuristic algorithm. Looking at the convergence curves obtained for the benchmark functions, it is clear that the proposed MIDA has a faster convergence capability than the original DA for the selected unimodal, multimodal and composition benchmark functions. It is shown that thanks to the effective mechanisms proposed in the structure of MIDA, the dragonflies in the swarm have fast and successful search performance in the exploration and exploitation stages while seeking the global solution point of the benchmark problem in the search space. It can also be seen from the convergence curves that, except for the FN1 benchmark function, the proposed MIDA has found the global solution to the other three benchmark functions (FN9, FN15, and FN25). Since the error values are zero for these three functions, the iteration is terminated so that there is no undefined situation in the logarithmic scale. This shows that MIDA reaches the real solution faster than the original DA structure with zero error. The convergence curve of the FN1 benchmark function, which has a more flat surface, shows that MIDA

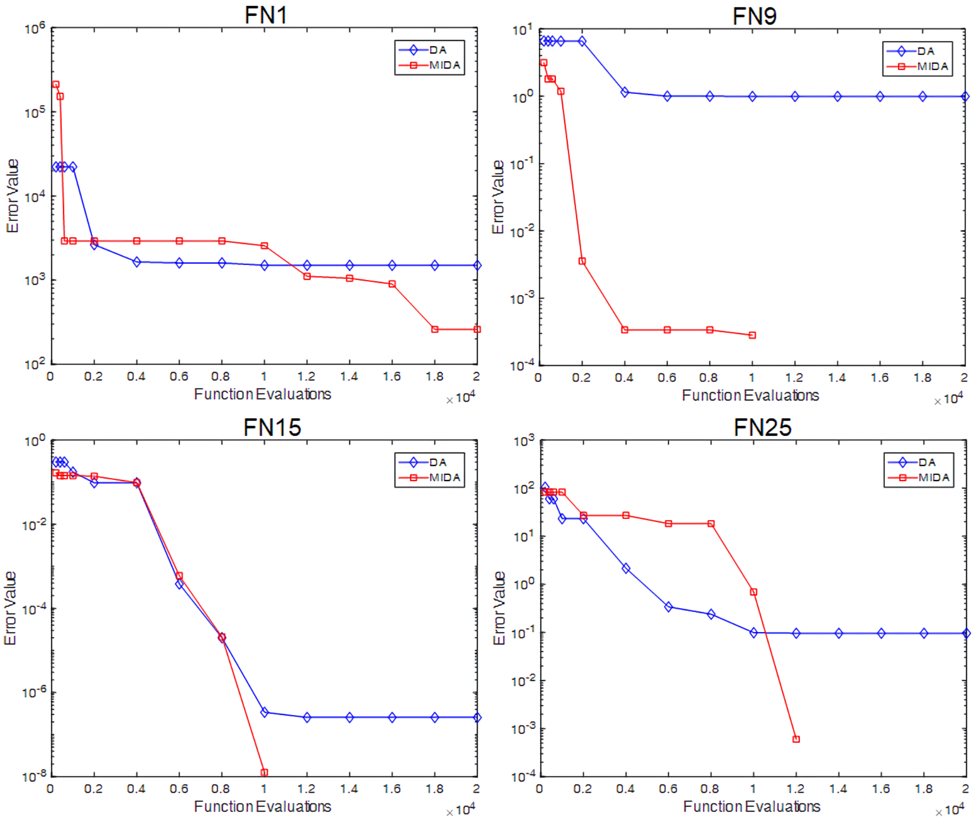


Figure 5: Convergence analysis results of the proposed MIDA and DA.

has a better convergence ability than the original DA in the exploitation phase, although worse than the convergence performance of DA in the exploration phase. At the end of the optimization for the FN1 function, MIDA has a lower error value than DA. The convergence analysis results demonstrate the capability of the MIDA to explore an answer faster and nearer to the global optimum. This is because of the mutation, boundary control, and greedy selection mechanisms proposed by the MIDA.

3.2. Search history analysis

The search history shows the position changes of dragonflies in the search space during the solution of MIDA's problem. This analysis provides information on how the positions of search agents (dragonflies) in the search space have changed. In Fig. 6, the search history results for the movements of the dragonfly swarm used in MIDA and DA are shown for selected benchmarks from the CEC2014 test suite. Search history analyzes were realized by taking into the same primary dragonfly positions for both algorithms. The updated positions of the dragonflies are indicated by different colors on the contour surfaces of the selected CEC2014 functions.

The search analysis reveals that the dragonfly positions round the global solution is greater than that of the positions updated by the DA during the exploration and exploitation stages of the optimization process. Because of the movements of dragonflies in the most promising areas of search space , it is clear that MIDA's search capability is high. Looking at the search history analysis results in Fig. 6, it is seen that there are some dragonflies that exceed the boundary values in the original DA, especially for the FN1, FN9, and FN25 functions, and they are located on the boundary value. This situation causes the dragonflies to get stuck at the boundary value while finding the global optimal point and reduces the success of the original DA in finding the global solution. Note that, thanks to the new boundary control mechanism proposed in the MIDA structure, there is no situation like this in MIDA's search analysis results.

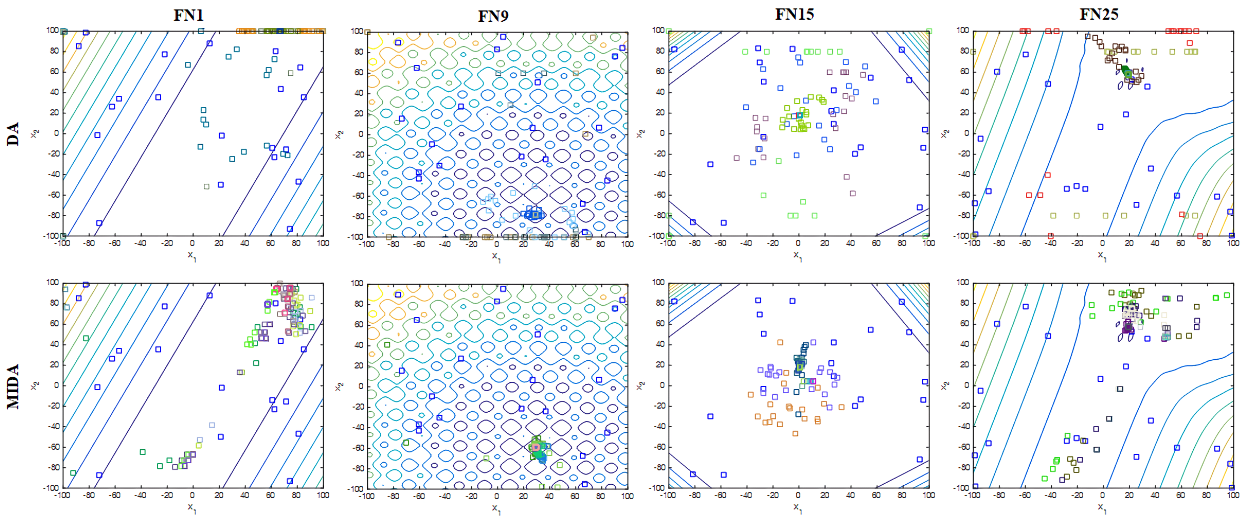


Figure 6: Search history analysis results of the MIDA and DA.

3.3. Trajectory analysis

In the trajectory analysis, while the global optima of the benchmark function are searched by the optimization algorithm, the changes in the position of the elite search agent (the best dragonfly) are examined. In this analysis, the initial positions of the dragonflies in both algorithms are the same. Two graphs are used in the trajectory analysis results (Fig. 7). The first (left) shows the position changes of the best dragonfly (elite) in the search space during optimization. In the first graph, ◊ represents the position of the elite dragonfly in DA, and □ represents the position of the elite dragonfly in MIDA. The positions of the elite dragonflies at the end of the optimization for both algorithms are shown in red. The second graph (right) gives the position changes of the elite dragonfly for both dimensions separately.

The trajectory analysis results of both algorithms for the FN1 benchmark function are shown in Fig. 7. FN1, which has a more flat surface, is a function of the unimodal group. It is seen from result of FN1, the position of the elite dragonfly found by DA gets stuck at a local minimum towards the end of the exploration phase whereas the position of the elite dragonfly found by MIDA changes during both phases. The most important reason for this is the mutation operator and greedy selection mechanism added to the original DA.

In Fig. 8, the FN9 benchmark function with the trajectory analysis results are shown for MIDA and DA structures. The results for final process show that the elite dragonfly positions obtained by both algorithms are not the same. The elite dragonfly position in MIDA is almost the global solution point of the FN9 function. In solving the optimization problem, in some cases, algorithms can find the global fitness value at different solution points. That is, while the f(x) value obtained at the end of the optimization represents the true fitness value of the problem, the solution point x may differ.This is more common in situations where the problem surface consists of many pits and hills like FN9.

Fig. 9 shows the trajectory analysis results of both algorithms for the FN15 benchmark. The results clearly show that MIDA's elite dragonfly approaches a global solution faster than DA's elite. Finally, the trajectory analysis results of the FN25 benchmark function are shown in Fig. 10. The position change of the elite dragonfly in the search space shows that both algorithms find almost the same global solution point, but MIDA reaches the solution point faster.

3.4. Average distance analysis

In this section, the variation of the distances between dragonflies in the swarm during optimization is examined. To do so, the average distance of the first dragonfly to the other dragonflies in the group during the optimization is taken according to the formula below:

$$\bar{d} = \frac{1}{N-1} \sum_{i=2}^{N} \left| X_{1,1} - X_{i,1} \right| \tag{17}$$

where $\bar{d}$ denotes the average distance, N represents the number of the dragonflies, $X_{1,1}$ is the first dimension of the first dragonfly, and $X_{i,1}$ is the first dimension of ith dragonfly.

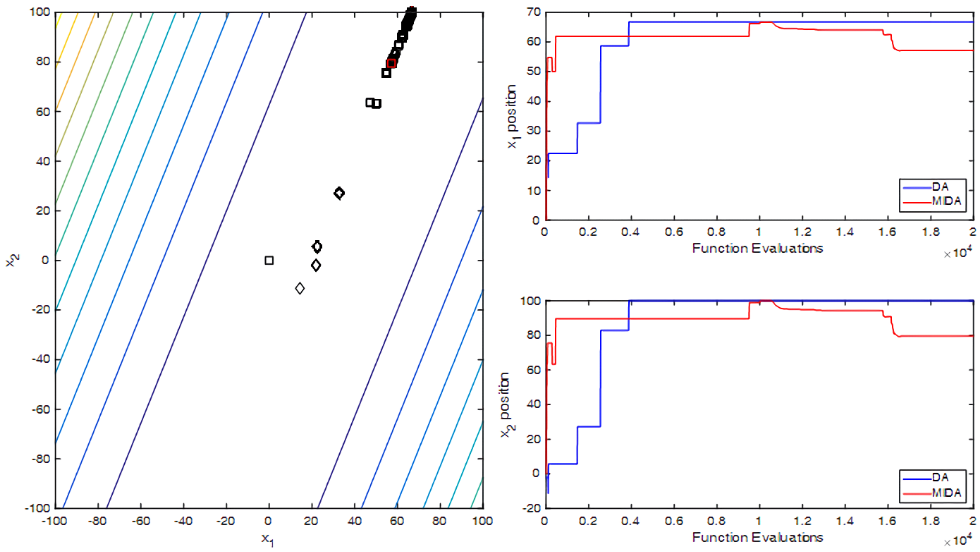


Figure 7: Trajectory of elite dragonfly for the FN1 benchmark function (◊: DA, □ : MIDA).

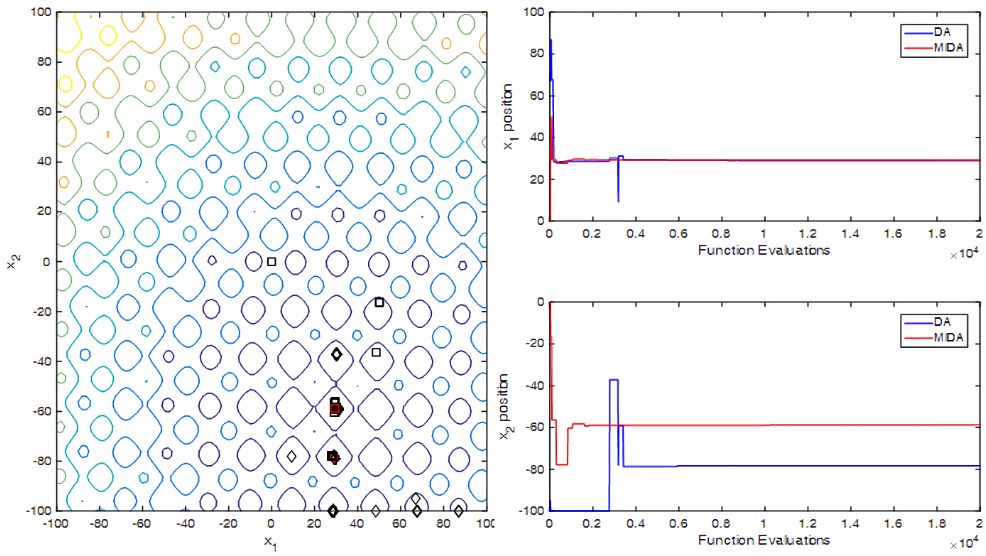


Figure 8: Trajectory of elite dragonfly for the FN9 benchmark function (◊: DA, □ : MIDA).

The results of the average distance analysis of MIDA and original DA for benchmark functions selected from CEC2014 test suite are shown in Fig. 11. It is clear that for the four benchmark functions, especially in the exploration phase, the average distance of the dragonfly swarm in the original DA has more fluctuations and oscillations than that of the dragonfly swarm of MIDA. This is because there is no selection method in the DA structure. However, thanks to the greedy selection mechanism added to the MIDA structure, it was observed that the oscillation and fluctuation decreased in the average distance analysis results of MIDA. When the analysis results of MIDA are examined, the peak values on the average distance curves formed in the exploitation phase show that the mutation operator implemented in the algorithm works.

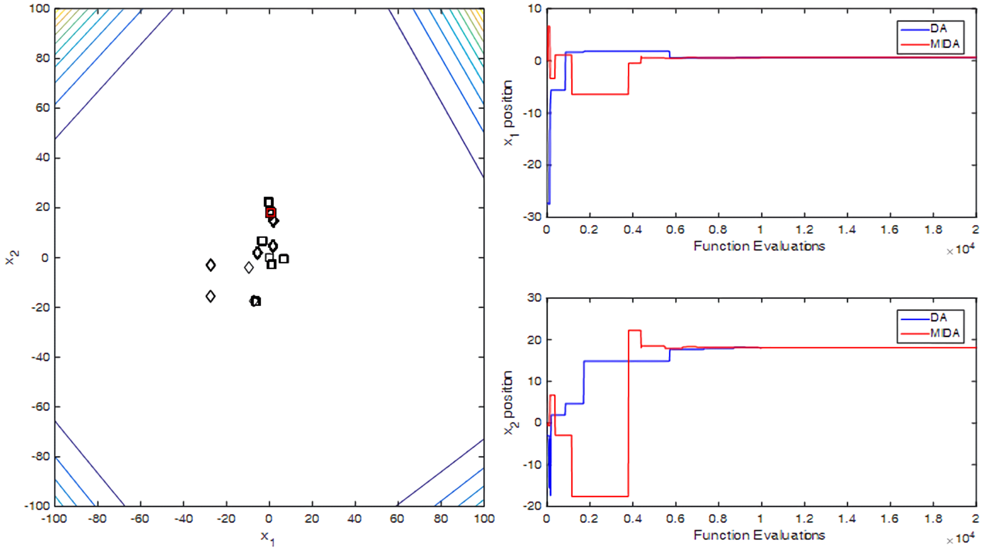


Figure 9: Trajectory of elite dragonfly for the FN15 benchmark function (◊: DA, □ : MIDA).

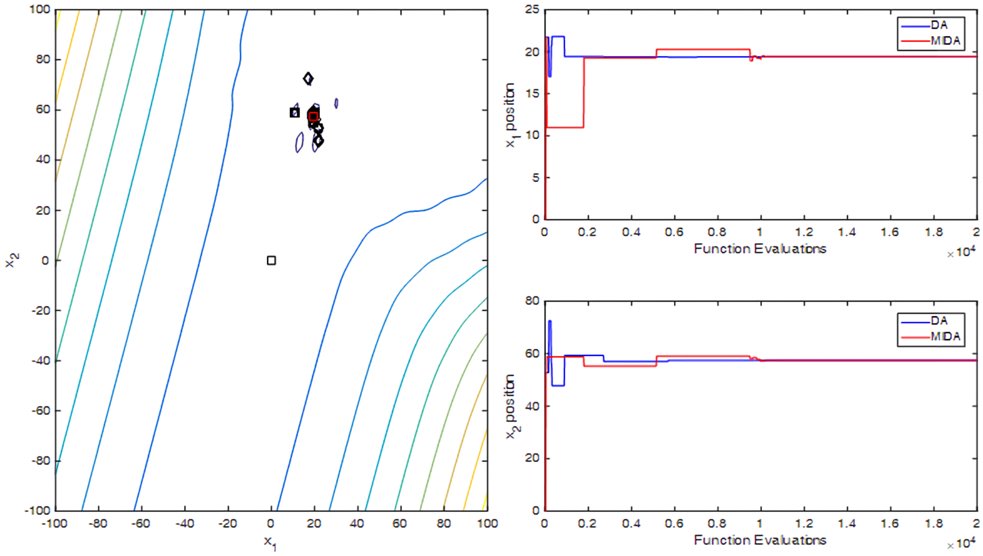


Figure 10: Trajectory of elite dragonfly for the FN25 benchmark function (◊: DA, □ : MIDA).

3.5. Computational complexity of MIDA

This section provides information on the computational complexity of the proposed MIDA.The computational complexity for the proposed MIDA was found as follows:

- For a population size of n and a problem dimension of d, the computational complexity for randomizing the positions of dragonflies is given in $O(n \times d)$.
- The fitness values for random positions of dragonflies are found, and the complexity of this process is $O(n \times d)$.
- In the main loop of the algorithm, the computational complexity is $O(t_m \times (2n^2 + 4n) \times d)$ for updating dragonflies' positions, mutation, boundary control, and greedy selection procedures. Here, t_m denotes the maximum iteration.

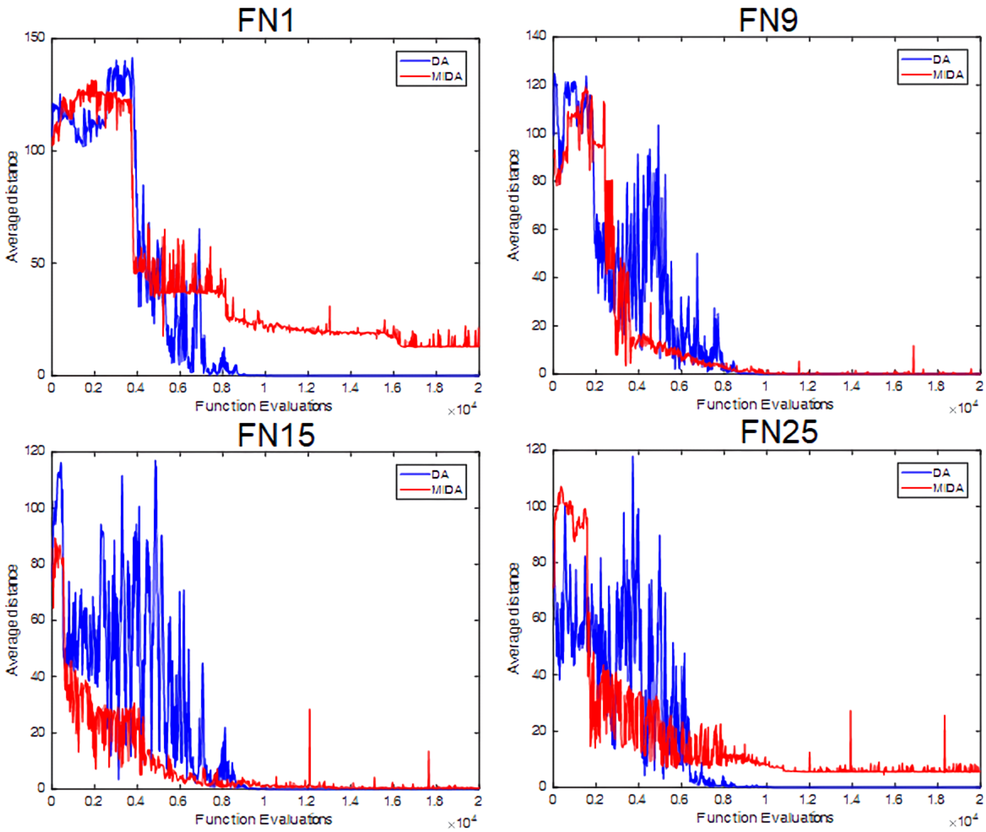


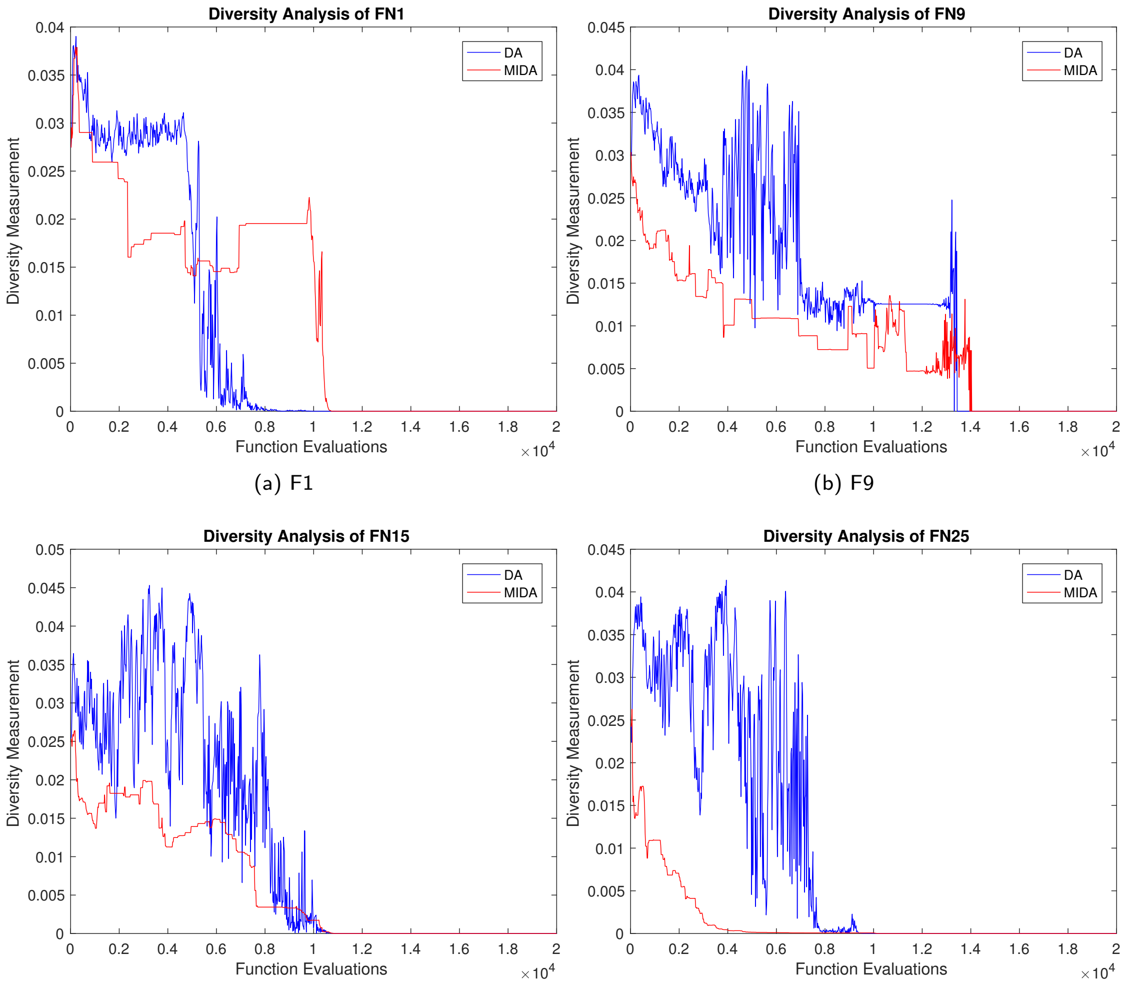
Figure 11: Average distance analysis of dragonflies

As a result, the total time complexity of MIDA is $O(2n \times d) + O(t_m \times (2n^2 + 4n) \times d)$ as the sum of the above expressions. From here, the computational complexity for MIDA is $O(t_m \times n^2 \times d)$ and is almost the same as the complexity of the original dragonfly algorithm.

3.6. Diversity analysis

In this section, diversity analyzes of both algorithms are performed. The diversity analysis results for F1, F9, F15, and F25 selected from the CEC2014 benchmarks are shown in Fig. 12. In the analysis, the diversity metric in the dragonfly population was normalized.

Initially, the diversity of both algorithms for each benchmark function is very high. This is because these algorithms start searching with random dragonfly positions. In the exploration and exploitation phases of the optimization process, the search regions of both algorithms are constantly narrowed, thus reducing the diversity in the dragonfly population. At the end of the optimization, the diversity curves for MIDA and DA were almost at the lowest level, indicating that the dragonflies in the population were in approximately the same place in the search space. However, decreasing trends in diversity curves are different for both algorithms. Except for F1, the MIDA's diversity curve for the other benchmarks decreases more rapidly than that of DA. In the F1 benchmark, the diversity curve of MIDA again showed a faster decrease than that of DA in the first 30% of the total iteration, but it is seen that the decrease in the curve of MIDA stopped in the next 20%. MIDA quickly recaptured the decrease in the diversity curve almost halfway through the optimization process and eventually reached the same result as DA, albeit with a slight delay. In conclusion, three mechanisms within the proposed MIDA have been demonstrated to have a visible effect on the diversity of the original DA.



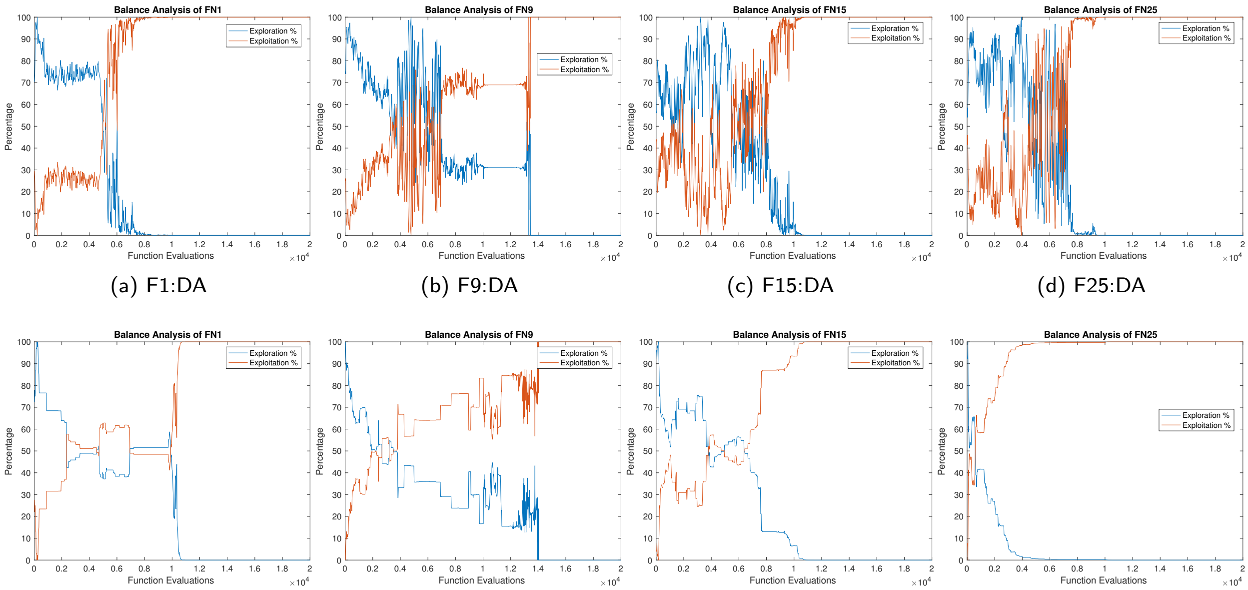
(a) F1 (b) F9

(c) F15 (d) F25

Figure 12: Diversity analysis results of the MIDA and DA.

3.7. Balance analysis

Balance analysis examines the search performance of meta-heuristic algorithms in terms of exploration and exploitation phases during optimization. In Fig. 13, the balance analysis results of MIDA and DA are shown for four benchmarks selected from the CEC2014 test suite. In each subfigure, the exploration and exploitation percentages throughout the optimization are shown in different colors. Larger values in the curves mean that the corresponding behavior (exploration or exploitation) dominates the algorithm. The curves obtained from the balance analysis results of the MIDA and DA show that the exploration behavior is dominant for both algorithms at the early stage. Although the balance analysis curves of DA have a lot of oscillations, it is generally understood that the exploration phase dominates in the early stage and the exploitation phase dominates the majority of the optimization process. MIDA's analysis results show that the dominant behavior pattern for other benchmarks, with the exception of F1, is exploitation. In addition, it is understood that there are no oscillations compared to the DA curves. This is due to the greedy selection mechanism in the MIDA structure. Looking at the results in general, it is clear that DA spends more time on exploration behavior than MIDA. However, the exploitative behavior of MIDA is more dominant, indicating that it spends more time in the search space to reach the solution. It is possible to say that the three different mechanisms (mutation operator, boundary control mechanism, and greedy selection mechanism) in the proposed MIDA structure cause the exploitation phase to have a greater effect on the search than the exploration and exploitation phases.



(a) F1:DA (b) F9:DA (c) F15:DA (d) F25:DA

(e) F1:MIDA (f) F9:MIDA (g) F15:MIDA (h) F25:MIDA

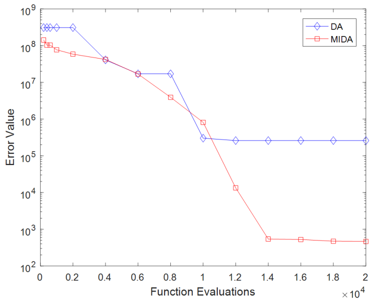
Figure 13: Balance analysis results of the MIDA and DA.

3.8. Comparison results for the CEC2014 benchmark problems

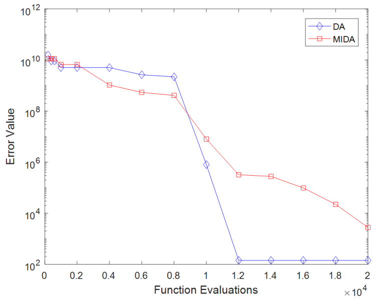
Statistical results for all benchmarks with different problem sizes (10D, 30D, 50D, and 100D) were found for fifty one iteration using the original DA and the MIDA. In solving the optimization problems, the parameters of both algorithms are taken to be the same. The size of the dragonfly swarm is 10 times the problem size, and the maximum number of iterations is 1000. The search termination criterion in the optimization process of both algorithms is to reach the maximum number of iterations. The codes of both algorithms were run on a PC with Intel(R) Core(TM) i7-6500U CPU@2.50 GHz with 8 GB RAM. A total of 14 error values were recorded at each run for each function in the CEC2014 test suite. Fig. 14 shows the best convergence curves of the F1 through the F15 benchmark functions with 10D for both algorithms. The best convergence curves obtained for the first 15 benchmark functions clearly show that MIDA achieves better results in 11 out of the 15 functions except for the F2, F5, F11, and F12. In Fig. 15, the best convergence curves obtained for the 10-dimensional F16-F30 comparison functions are shown. It can be seen that DA converges to the global solution better than MIDA for the four benchmarks (F23, F27, F28, and F29) while the proposed MIDA has a better convergence for the remaining convergence curves. In summary, MIDA is more successful in 22 out of 30 functions, while DA is only successful in 8 out of 30 functions, according to the best convergence curves presented for 10-dimensional benchmark functions.

In order to better understand the comparison results between the MIDA and DA, statistical metrics were calculated from the results using 51 runs of each benchmark function. In the comparison results of both algorithms, five metrics were evaluated, such as mean, worst, best, median, and standard deviation. The statistical results of both algorithms for the 10D and 30D CEC2014 benchmarks are summarised in Table 2 and Table 3, respectively. From the tables, the best results across all metrics are highlighted in bold. Looking at the table results obtained for the 10D benchmark functions, it is clearly evident that MIDA has better results than the original DA by 73.33% for the best metric, 100.00% for the worst metric, 93.33% for the median metric, 93.33% for the mean metric, and 86.67% for the standard deviation metric. In the statistical results for the 30-dimensional CEC2014 benchmark functions, it is seen that MIDA has better results than those of the original DA by 80.00% for the best metric, 96.67% for the worst metric, 93.33% for the median metric, 93.33% for the mean metric, and 93.33% for the standard deviation metric.

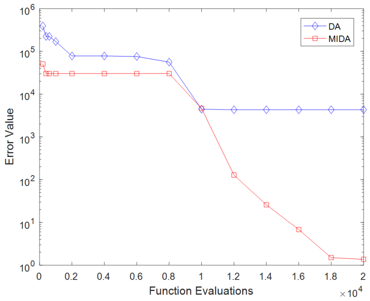
Table 4 and Table 5 summarizes the statistical results of MIDA and DA for CEC2014 benchmarks with 50D and 100D. The 50-dimensional test problem results show that MIDA's performance is better than that of the original DA by 86.67% for the best metric, 96.67% for the worst metric, 90.00% for the median metric, 90.00% for the average metric, and 93.33% for the standard deviation metric. From the optimization results of 100D benchmarks, The proposed MIDA



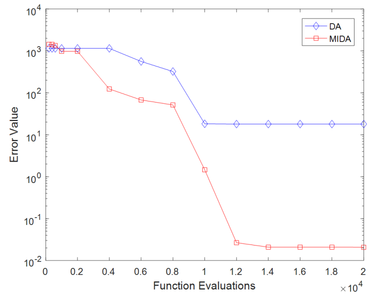
(a) F1



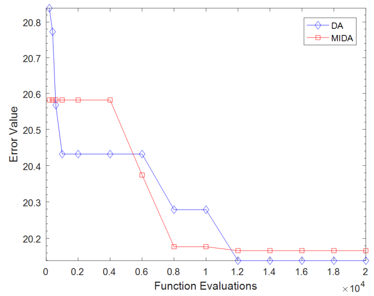
(b) F2



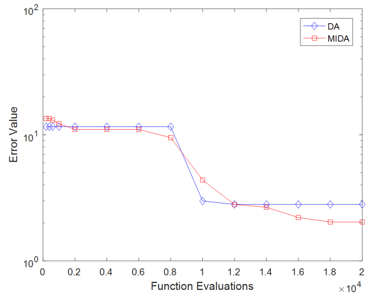
(c) F3



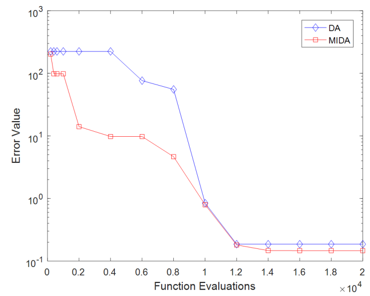
(d) F4



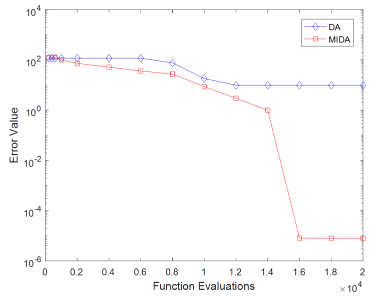
(e) F5



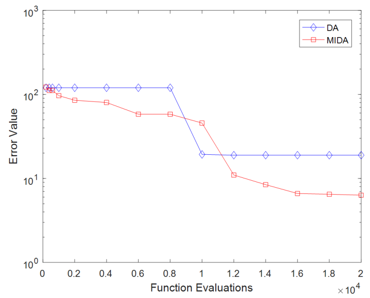
(f) F6



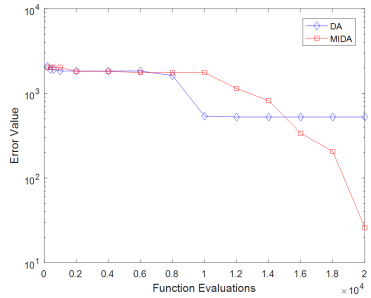
(g) F7



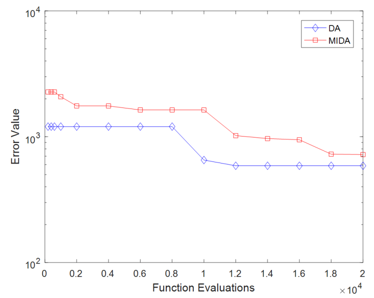
(h) F8



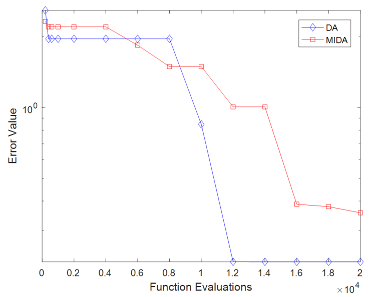
(i) F9



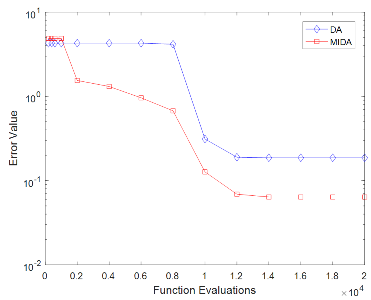
(j) F10



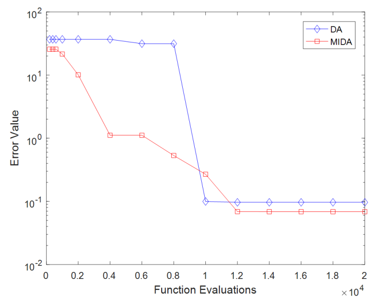
(k) F11



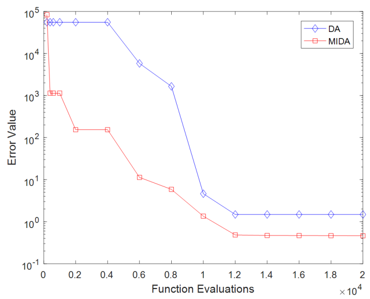
(l) F12



(m) F13

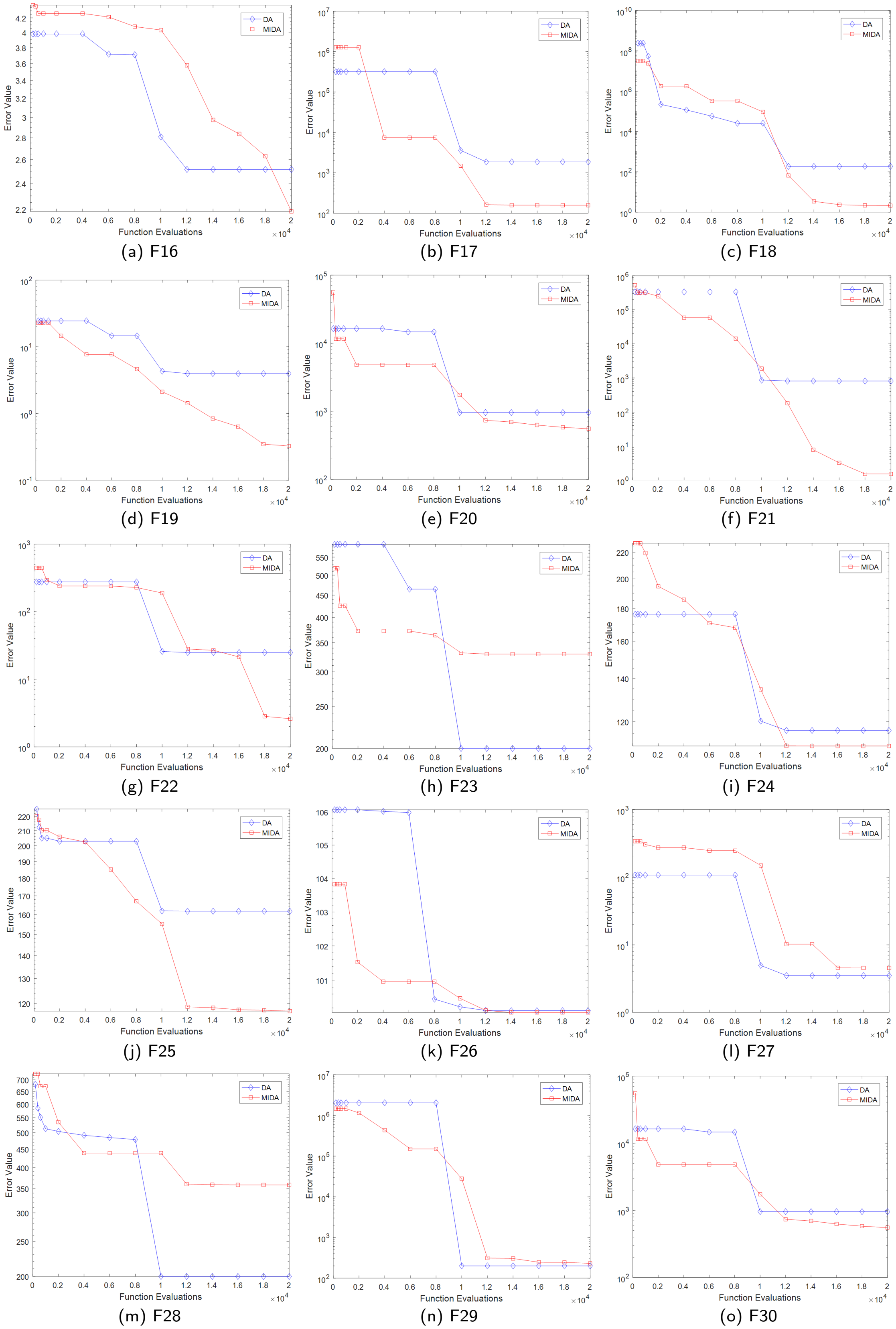


(n) F14



(o) F15

Figure 14: Convergence curves of the MIDA and DA for F1-F15 functions.



(a) F16 (b) F17 (c) F18
(d) F19 (e) F20 (f) F21
(g) F22 (h) F23 (i) F24
(j) F25 (k) F26 (l) F27
(m) F28 (n) F29 (o) F30

Figure 15: Convergence curves of the MIDA and DA for F16-F30 functions.

Table 2
10D CEC2014 Benchmark Results for DA and MIDA.

	DA					MIDA				
No	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
1	2.60E+5	1.31E+8	1.27E+7	2.71E+7	3.61E+7	**4.65E+2**	**1.98E+6**	**1.23E+5**	**1.52E+5**	**2.76E+5**
2	**1.42E+2**	5.21E+9	2.56E+8	1.04E+9	1.31E+9	2.78E+3	**2.38E+5**	**3.00E+4**	**4.38E+4**	**4.61E+4**
3	4.31E+3	1.16E+5	3.06E+4	3.61E+4	2.79E+4	**1.37E+0**	**1.30E+4**	**1.11E+3**	**2.06E+3**	**3.15E+3**
4	1.80E+1	1.18E+3	7.79E+1	1.33E+2	2.03E+2	**2.07E-2**	**3.64E+1**	**3.49E+1**	**2.10E+1**	**1.66E+1**
5	**2.01E+1**	2.07E+1	**2.03E+1**	**2.03E+1**	1.44E-1	2.02E+1	**2.05E+1**	2.04E+1	2.04E+1	**8.17E-2**
6	2.81E+0	1.19E+1	9.01E+0	8.79E+0	**1.71E+0**	**2.03E+0**	**9.57E+0**	**6.59E+0**	**6.30E+0**	1.89E+0
7	1.86E-1	1.61E+2	1.90E+1	2.32E+1	3.24E+1	**1.47E-1**	**1.05E+0**	**5.12E-1**	**5.25E-1**	**2.18E-1**
8	9.81E+0	9.95E+1	4.71E+1	4.67E+1	1.84E+1	**8.02E-6**	**5.65E-2**	**5.44E-3**	**1.04E-2**	**1.34E-2**
9	1.89E+1	8.26E+1	4.60E+1	4.85E+1	1.86E+1	**6.32E+0**	**4.58E+1**	**1.70E+1**	**2.01E+1**	**9.04E+0**
10	5.25E+2	1.63E+3	1.11E+3	1.07E+3	2.51E+2	**2.59E+1**	**7.68E+2**	**2.61E+2**	**2.88E+2**	**1.85E+2**
11	**5.88E+2**	1.99E+3	1.45E+3	1.42E+3	**3.15E+2**	7.21E+2	**1.90E+3**	**1.45E+3**	**1.37E+3**	3.17E+2
12	**2.21E-1**	2.14E+0	**9.99E-1**	**9.93E-1**	4.45E-1	3.57E-1	**1.52E+0**	1.06E+0	1.07E+0	**3.07E-1**
13	1.87E-1	3.80E+0	6.01E-1	1.20E+0	1.10E+0	**6.39E-2**	**7.45E-1**	**2.65E-1**	**3.17E-1**	**1.58E-1**
14	9.66E-2	1.79E+1	5.13E+0	5.10E+0	4.41E+0	**6.87E-2**	**9.75E-1**	**2.46E-1**	**3.00E-1**	**2.14E-1**
15	1.48E+0	5.82E+4	2.56E+1	1.73E+3	8.23E+3	**4.62E-1**	**5.35E+0**	**1.80E+0**	**1.86E+0**	**8.69E-1**
16	2.52E+0	4.18E+0	3.86E+0	3.75E+0	**3.71E-1**	**2.18E+0**	**3.75E+0**	**3.07E+0**	**3.00E+0**	3.80E-1
17	1.87E+3	3.24E+6	4.39E+4	3.67E+5	8.34E+5	**1.58E+2**	**4.17E+4**	**1.23E+3**	**4.08E+3**	**6.87E+3**
18	1.86E+2	3.80E+4	1.59E+3	1.58E+4	1.72E+4	**2.11E+0**	**9.53E+3**	**4.28E+1**	**7.21E+2**	**1.76E+3**
19	3.96E+0	1.23E+1	7.90E+0	7.76E+0	1.43E+0	**3.25E-1**	**3.88E+0**	**1.80E+0**	**2.13E+0**	**9.29E-1**
20	2.22E+2	2.12E+5	5.38E+3	2.02E+4	4.20E+4	**1.65E+0**	**2.72E+3**	**1.07E+1**	**7.34E+1**	**3.80E+2**
21	8.12E+2	3.53E+6	3.19E+3	1.43E+5	6.80E+5	**1.52E+0**	**2.83E+3**	**1.42E+2**	**4.15E+2**	**6.07E+2**
22	2.49E+1	4.09E+2	2.33E+2	2.15E+2	1.10E+2	**2.60E+0**	**2.12E+2**	**2.12E+1**	**3.26E+1**	**3.73E+1**
23	**2.00E+2**	4.43E+2	3.65E+2	3.53E+2	5.47E+1	3.29E+2	**3.30E+2**	**3.29E+2**	**3.29E+2**	**3.13E-2**
24	1.16E+2	2.23E+2	1.74E+2	1.73E+2	2.48E+1	**1.10E+2**	**1.53E+2**	**1.29E+2**	**1.31E+2**	**1.03E+1**
25	1.62E+2	2.09E+2	2.03E+2	2.02E+2	**7.63E+0**	**1.17E+2**	**2.03E+2**	**1.74E+2**	**1.72E+2**	3.16E+1
26	1.00E+2	1.02E+2	1.01E+2	1.01E+2	4.46E-1	**1.00E+2**	**1.01E+2**	**1.00E+2**	**1.00E+2**	**2.22E-1**
27	**3.51E+0**	6.37E+2	4.18E+2	3.73E+2	2.04E+2	4.54E+0	**4.03E+2**	**4.00E+2**	**2.31E+2**	**1.95E+2**
28	**2.00E+2**	7.82E+2	4.68E+2	4.99E+2	1.24E+2	3.58E+2	**5.47E+2**	**4.01E+2**	**4.21E+2**	**5.14E+1**
29	**2.00E+2**	1.96E+6	2.54E+3	1.16E+5	4.47E+5	2.30E+2	**1.10E+3**	**5.55E+2**	**5.82E+2**	**2.25E+2**
30	9.57E+2	5.24E+4	3.34E+3	5.61E+3	9.17E+3	**5.54E+2**	**2.74E+3**	**8.11E+2**	**1.01E+3**	**4.57E+2**

is better than DA in 83.33% of all test problems for the best metric values, better than DA in 100% of all test problems for the worst metric values, and DA in 96.67% of all test problems for mean values. It has better results than DA in 93.33% of all test problems for median values, better than DA in 93.33% of all test problems for standard deviation values. As a result, the proposed MIDA is found to be more successful than the DA structure in terms of all statistical metrics, with 89.33% for 10D, 91.33% for 30D, 91.33% for 50D, and 93.33% for 100D in the optimization of CEC2014 benchmark functions.

The results of the proposed MIDA and the original DA were compared using the Wilcoxon rank-sum test, that is one of the non-parametric tests for statistical significance for all problem dimensions. Usually, this test is a statistical method often preferred to compare repeated measurements on a single sample or to compare two samples. Table 6 presents the p-value scores obtained by Wilcoxon rank-sum test with 5% accuracy from a pair of samples for 51 independent runs of MIDA and DA structures to test the null hypothesis. A confidence level of 0.95 was used for statistical analysis in

Table 3
30D CEC2014 Benchmark Results for DA and MIDA.

	DA					MIDA				
No	**Best**	**Worst**	**Median**	**Mean**	**Std**	**Best**	**Worst**	**Median**	**Mean**	**Std**
1	6.79E+7	1.93E+9	3.35E+8	4.49E+8	3.88E+8	**2.51E+6**	**2.95E+7**	**8.72E+6**	**1.07E+7**	**6.70E+6**
2	8.81E+9	7.00E+10	3.83E+10	3.67E+10	1.55E+10	**2.17E+7**	**1.51E+8**	**4.63E+7**	**4.93E+7**	**2.10E+7**
3	6.34E+4	4.91E+5	1.64E+5	1.81E+5	9.61E+4	**7.12E+1**	**1.13E+4**	**1.71E+3**	**2.50E+3**	**2.63E+3**
4	3.15E+2	1.55E+4	3.43E+3	4.86E+3	3.72E+3	**9.38E+1**	**3.19E+2**	**1.59E+2**	**1.71E+2**	**5.04E+1**
5	**2.05E+1**	**2.10E+1**	**2.09E+1**	2.08E+1	1.17E-1	2.08E+1	2.11E+1	2.09E+1	2.09E+1	**7.07E-2**
6	2.81E+1	4.37E+1	3.77E+1	3.72E+1	**3.71E+0**	**1.50E+1**	**3.49E+1**	**2.38E+1**	**2.46E+1**	4.38E+0
7	8.18E+1	7.37E+2	3.74E+2	3.54E+2	1.59E+2	**1.21E+0**	**1.81E+0**	**1.42E+0**	**1.45E+0**	**1.44E-1**
8	1.82E+2	3.74E+2	2.89E+2	2.89E+2	4.77E+1	**1.31E+1**	**3.66E+1**	**2.03E+1**	**2.07E+1**	**4.59E+0**
9	1.55E+2	4.30E+2	2.97E+2	3.01E+2	6.73E+1	**6.02E+1**	**2.06E+2**	**1.13E+2**	**1.15E+2**	**3.14E+1**
10	4.82E+3	8.10E+3	6.67E+3	6.57E+3	6.79E+2	**9.62E+1**	**2.41E+3**	**1.16E+3**	**1.28E+3**	**5.48E+2**
11	**4.93E+3**	8.17E+3	**7.10E+3**	6.92E+3	7.41E+2	5.39E+3	**7.86E+3**	7.16E+3	7.03E+3	**5.53E+2**
12	**9.11E-1**	3.18E+0	2.26E+0	2.21E+0	5.26E-1	9.66E-1	**3.12E+0**	**2.14E+0**	**2.19E+0**	**4.75E-1**
13	5.77E-1	7.99E+0	4.96E+0	5.14E+0	1.65E+0	**3.40E-1**	**8.27E-1**	**5.46E-1**	**5.57E-1**	**1.15E-1**
14	4.66E+1	2.84E+2	1.20E+2	1.31E+2	5.52E+1	**1.75E-1**	**1.03E+0**	**3.10E-1**	**3.34E-1**	**1.45E-1**
15	1.68E+3	2.21E+6	1.40E+5	2.85E+5	4.25E+5	**1.36E+1**	**3.53E+1**	**2.24E+1**	**2.31E+1**	**4.90E+0**
16	1.23E+1	1.36E+1	1.32E+1	1.32E+1	**3.23E-1**	**1.09E+1**	**1.32E+1**	**1.24E+1**	**1.23E+1**	5.70E-1
17	2.02E+6	6.45E+7	1.23E+7	1.61E+7	1.36E+7	**1.74E+4**	**2.83E+6**	**2.97E+5**	**5.35E+5**	**6.14E+5**
18	4.10E+4	3.04E+9	2.58E+8	6.11E+8	8.31E+8	**8.34E+1**	**1.06E+5**	**1.00E+4**	**1.65E+4**	**1.83E+4**
19	7.89E+1	5.20E+2	2.51E+2	2.68E+2	1.08E+2	**7.75E+0**	**8.49E+1**	**1.25E+1**	**2.12E+1**	**2.27E+1**
20	2.18E+4	3.48E+6	1.06E+5	2.78E+5	5.77E+5	**1.57E+2**	**3.03E+3**	**5.46E+2**	**8.13E+2**	**7.12E+2**
21	2.38E+5	3.23E+7	4.21E+6	8.23E+6	8.77E+6	**5.69E+3**	**6.16E+5**	**5.07E+4**	**8.45E+4**	**1.01E+5**
22	4.50E+2	2.16E+3	1.12E+3	1.12E+3	4.04E+2	**4.38E+1**	**8.49E+2**	**4.35E+2**	**4.39E+2**	**1.97E+2**
23	**2.00E+2**	1.25E+3	5.39E+2	5.76E+2	1.98E+2	3.15E+2	**3.17E+2**	**3.16E+2**	**3.16E+2**	**3.66E-1**
24	**2.04E+2**	4.19E+2	2.78E+2	2.81E+2	4.14E+1	2.27E+2	**2.45E+2**	**2.33E+2**	**2.34E+2**	**4.51E+0**
25	**2.00E+2**	3.18E+2	2.36E+2	2.40E+2	2.28E+1	2.05E+2	**2.17E+2**	**2.10E+2**	**2.10E+2**	**3.23E+0**
26	1.00E+2	1.08E+2	1.04E+2	1.04E+2	1.69E+0	**1.00E+2**	**1.01E+2**	**1.01E+2**	**1.01E+2**	**1.35E-1**
27	4.50E+2	1.64E+3	1.24E+3	1.09E+3	3.58E+2	**4.04E+2**	**9.63E+2**	**4.08E+2**	**4.93E+2**	**1.75E+2**
28	1.40E+3	3.77E+3	2.29E+3	2.24E+3	5.47E+2	**9.21E+2**	**2.49E+3**	**1.54E+3**	**1.59E+3**	**4.08E+2**
29	7.97E+4	6.87E+7	1.92E+7	2.15E+7	1.09E+7	**1.25E+3**	**2.36E+7**	**1.79E+3**	**1.64E+6**	**5.19E+6**
30	5.83E+4	1.63E+6	3.20E+5	4.79E+5	3.77E+5	**2.14E+3**	**5.05E+4**	**7.97E+3**	**1.14E+4**	**1.01E+4**

this table, and p values less than 0.05 are shown in bold. Wilcoxon rank-sum test results show that there are significant differences between the repetitive optimization results, obtained by the original DA and the proposed MIDA, for all benchmark problems in four different dimensions. It is only understood that there is not a very significant difference in the F12 results for all problem dimensions. Similarly, statistically, there is no significant difference between the results of both algorithms for F15 in all dimensions except 30D and for F11 in all dimensions except 50D.

Secondly, the proposed MIDA was statistically compared with several state-of-the-art meta-heuristic algorithms for the 30-dimensional CEC2014 benchmark functions. The meta-heuristic algorithms considered are Particle Swarm Optimizer (PSO) [8], Moth-Flame Optimizer [74], Sine Cosine Algorithm (SCA) [75], Dragonfly Algorithm (DA) [28], Whale Optimization Algorithm (WOA) [76], Chaotic Cuckoo Search optimizer (CCS) [77], Coyote Optimization Algorithm (COA) [78], Elephant Herding Optimizer (EHO) [79], Ageist Spider Monkey Optimizer (ASMO) [80], Variable Neighborhood Bat Algorithm (VNBA) [81], Red Fox Optimizer (RFO) [82], and Human Felicity Algorithm

Table 4
50D CEC2014 Benchmark Results for DA and MIDA.

	DA					MIDA				
No	**Best**	**Worst**	**Median**	**Mean**	**Std**	**Best**	**Worst**	**Median**	**Mean**	**Std**
1	7.56E+8	2.73E+9	1.27E+9	1.43E+9	6.96E+8	**4.48E+7**	**1.17E+8**	**6.79E+7**	**6.94E+7**	**2.28E+7**
2	1.92E+10	1.38E+11	4.95E+10	5.61E+10	3.03E+10	**8.75E+8**	**1.49E+9**	**1.16E+9**	**1.18E+9**	**2.15E+8**
3	2.07E+5	4.83E+5	3.25E+5	3.20E+5	8.44E+4	**1.22E+4**	**5.00E+4**	**2.36E+4**	**2.39E+4**	**9.84E+3**
4	2.97E+3	1.78E+4	8.61E+3	8.51E+3	4.16E+3	**2.83E+2**	**6.32E+2**	**4.28E+2**	**4.30E+2**	**1.00E+2**
5	**2.09E+1**	**2.13E+1**	**2.12E+1**	2.11E+1	1.49E-1	2.11E+1	2.12E+1	2.12E+1	2.12E+1	**5.16E-2**
6	6.11E+1	7.70E+1	7.04E+1	6.92E+1	**5.71E+0**	**4.89E+1**	**6.71E+1**	**6.03E+1**	**6.05E+1**	4.80E+0
7	2.56E+2	1.26E+3	5.39E+2	6.56E+2	3.39E+2	**8.73E+0**	**1.54E+1**	**1.19E+1**	**1.21E+1**	**1.82E+0**
8	4.58E+2	6.84E+2	5.39E+2	5.63E+2	7.50E+1	**1.73E+2**	**2.63E+2**	**2.18E+2**	**2.21E+2**	**2.60E+1**
9	3.90E+2	8.19E+2	7.00E+2	6.57E+2	1.25E+2	**4.06E+2**	**5.59E+2**	**4.67E+2**	**4.65E+2**	**4.55E+1**
10	1.13E+4	1.50E+4	1.38E+4	1.35E+4	1.20E+3	**4.98E+3**	**8.11E+3**	**6.15E+3**	**6.32E+3**	**1.05E+3**
11	**9.62E+3**	1.46E+4	**1.32E+4**	1.28E+4	1.44E+3	1.33E+4	**1.48E+4**	1.39E+4	1.39E+4	**3.96E+2**
12	**1.75E+0**	4.33E+0	3.42E+0	3.43E+0	7.77E-1	2.87E+0	**4.14E+0**	**3.48E+0**	**3.49E+0**	**3.80E-1**
13	3.35E+0	6.97E+0	4.40E+0	4.87E+0	1.17E+0	**3.41E-1**	**8.37E-1**	**6.27E-1**	**6.25E-1**	**1.46E-1**
14	5.88E+1	2.35E+2	1.26E+2	1.33E+2	5.06E+1	**3.06E-1**	**8.72E-1**	**3.39E-1**	**4.63E-1**	**2.22E-1**
15	1.17E+5	2.21E+6	2.45E+5	5.87E+5	6.74E+5	**6.52E+1**	**1.68E+2**	**9.81E+1**	**1.05E+2**	**3.24E+1**
16	2.24E+1	2.38E+1	2.30E+1	2.31E+1	**4.16E-1**	**2.20E+1**	**2.34E+1**	**2.28E+1**	**2.28E+1**	4.75E-1
17	2.88E+7	1.63E+8	6.81E+7	9.07E+7	4.46E+7	**2.35E+6**	**1.91E+7**	**8.56E+6**	**9.66E+6**	**5.48E+6**
18	1.11E+8	3.81E+9	5.33E+8	9.60E+8	1.09E+9	**6.45E+4**	**3.17E+6**	**1.86E+5**	**5.56E+5**	**9.09E+5**
19	1.17E+2	8.20E+2	3.55E+2	3.99E+2	2.06E+2	**3.38E+1**	**9.65E+1**	**7.75E+1**	**7.27E+1**	**2.15E+1**
20	8.16E+4	1.63E+6	3.27E+5	5.16E+5	5.11E+5	**6.14E+3**	**3.87E+4**	**2.26E+4**	**2.40E+4**	**1.15E+4**
21	6.72E+6	7.82E+7	4.12E+7	4.02E+7	2.66E+7	**5.88E+5**	**5.63E+6**	**2.89E+6**	**3.13E+6**	**1.51E+6**
22	1.81E+3	3.65E+3	2.43E+3	2.63E+3	5.92E+2	**7.90E+2**	**2.05E+3**	**1.30E+3**	**1.29E+3**	**3.64E+2**
23	**4.90E+2**	1.16E+3	8.01E+2	8.09E+2	1.84E+2	3.47E+2	**3.55E+2**	**3.50E+2**	**3.50E+2**	**2.44E+0**
24	**3.07E+2**	5.09E+2	3.99E+2	4.17E+2	6.06E+1	2.84E+2	**3.06E+2**	**2.98E+2**	**2.97E+2**	**6.80E+0**
25	**2.73E+2**	3.56E+2	3.10E+2	3.08E+2	2.30E+1	2.26E+2	**2.50E+2**	**2.39E+2**	**2.40E+2**	**7.71E+0**
26	1.04E+2	4.79E+2	1.05E+2	1.40E+2	1.12E+2	**1.00E+2**	**2.07E+2**	**1.01E+2**	**1.38E+2**	**5.23E+1**
27	1.95E+3	2.54E+3	2.20E+3	2.19E+3	1.56E+2	**1.61E+3**	**1.93E+3**	**1.82E+3**	**1.78E+3**	**1.03E+2**
28	4.68E+3	8.18E+3	5.90E+3	6.01E+3	1.09E+3	**4.39E+3**	**6.28E+3**	**5.16E+3**	**5.25E+3**	**5.83E+2**
29	1.13E+8	4.26E+8	3.24E+8	2.97E+8	9.29E+7	**3.95E+5**	**3.85E+8**	**1.75E+8**	**1.42E+8**	**1.27E+8**
30	1.10E+6	4.73E+7	4.73E+6	8.94E+6	1.31E+7	**2.82E+4**	**1.99E+5**	**5.65E+4**	**6.82E+4**	**4.76E+4**

(HFA) [83] . The parameter values of the selected meta-heuristic algorithms were taken as in the original articles. Table 7 and Table 8 give the comparison results between the proposed MIDA and the other twelve meta-heuristic algorithms for the 30D CEC2014 benchmarks. This table contains the mean and standard deviation metrics for 51 runs of all algorithms. The last row of each comparison function is the ranking of the algorithms according to their mean error values. The average rank in the table refers to the average of the rankings obtained for all benchmark functions of each algorithm while the overall rank is the average rank values of the MIDA and the meta-heuristic algorithms considered. As can be seen from both tables, the proposed MIDA for the 30-dimensional CEC2014 benchmark functions outperforms other implemented algorithms. The overall ranking obtained from the average rankings in the last row of the tables clearly shows that MIDA is the first algorithm among the other algorithms. With the effect of three different mechanisms applied to the original dragonfly algorithm, MIDA can be said to have a more effective

Table 5
100D CEC2014 Benchmark Results for DA and MIDA.

	DA					MIDA				
No	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
1	1.91E+9	8.55E+9	3.61E+9	4.13E+9	1.91E+9	**2.16E+8**	**4.78E+8**	**3.74E+8**	**3.70E+8**	**8.57E+7**
2	1.27E+11	2.59E+11	1.69E+11	1.75E+11	3.97E+10	**4.45E+9**	**7.17E+9**	**5.52E+9**	**5.59E+9**	**7.83E+8**
3	3.33E+5	1.14E+6	7.09E+5	7.09E+5	2.26E+5	**1.77E+4**	**4.61E+4**	**2.86E+4**	**3.05E+4**	**8.85E+3**
4	1.85E+4	5.79E+4	4.39E+4	3.89E+4	1.51E+4	**8.77E+2**	**1.44E+3**	**1.08E+3**	**1.12E+3**	**2.21E+2**
5	**2.10E+1**	2.14E+1	**2.13E+1**	**2.13E+1**	1.50E-1	2.13E+1	**2.14E+1**	2.14E+1	2.13E+1	**3.04E-2**
6	1.47E+2	1.69E+2	1.53E+2	1.55E+2	**6.66E+0**	**1.21E+2**	**1.45E+2**	**1.30E+2**	**1.30E+2**	7.22E+0
7	1.33E+3	2.88E+3	1.99E+3	1.99E+3	5.52E+2	**4.05E+1**	**5.66E+1**	**4.77E+1**	**4.84E+1**	**4.54E+0**
8	1.25E+3	1.48E+3	1.36E+3	1.36E+3	7.85E+1	**4.58E+2**	**6.88E+2**	**6.11E+2**	**5.98E+2**	**5.81E+1**
9	1.16E+3	1.62E+3	1.47E+3	1.45E+3	1.37E+2	**1.10E+3**	**1.41E+3**	**1.30E+3**	**1.28E+3**	**9.31E+1**
10	2.54E+4	3.38E+4	3.00E+4	3.01E+4	2.35E+3	**1.31E+4**	**1.88E+4**	**1.54E+4**	**1.52E+4**	**1.63E+3**
11	2.89E+4	3.33E+4	3.06E+4	3.10E+4	**1.44E+3**	**2.64E+4**	**3.12E+4**	**2.98E+4**	**2.93E+4**	1.75E+3
12	3.52E+0	4.66E+0	4.09E+0	4.04E+0	3.48E-1	**3.36E+0**	**4.49E+0**	**3.99E+0**	**4.00E+0**	**3.47E-1**
13	5.92E+0	8.72E+0	7.93E+0	7.57E+0	9.61E-1	**4.84E-1**	**7.91E-1**	**6.43E-1**	**6.38E-1**	**9.24E-2**
14	3.95E+2	7.17E+2	5.94E+2	6.00E+2	1.09E+2	**2.86E-1**	**7.33E+0**	**3.58E-1**	**9.89E-1**	**2.10E+0**
15	2.28E+6	1.18E+7	4.85E+6	5.88E+6	3.25E+6	**3.43E+2**	**1.32E+3**	**6.90E+2**	**7.54E+2**	**3.40E+2**
16	**4.63E+1**	4.81E+1	4.76E+1	4.75E+1	5.22E-1	4.63E+1	**4.74E+1**	**4.69E+1**	**4.68E+1**	**3.88E-1**
17	2.99E+8	8.54E+8	4.98E+8	5.14E+8	1.74E+8	**8.67E+6**	**5.91E+7**	**3.36E+7**	**3.33E+7**	**1.59E+7**
18	6.44E+8	1.77E+10	7.41E+9	9.10E+9	6.70E+9	**3.24E+5**	**5.55E+6**	**8.28E+5**	**1.35E+6**	**1.48E+6**
19	7.46E+2	2.75E+3	1.55E+3	1.61E+3	6.07E+2	**1.57E+2**	**2.28E+2**	**1.77E+2**	**1.83E+2**	**2.25E+1**
20	3.49E+5	4.02E+6	1.93E+6	2.00E+6	1.44E+6	**1.26E+4**	**4.96E+4**	**2.68E+4**	**2.74E+4**	**9.14E+3**
21	6.40E+7	4.51E+8	1.92E+8	1.92E+8	1.11E+8	**7.06E+6**	**3.15E+7**	**1.68E+7**	**1.61E+7**	**6.64E+6**
22	3.43E+3	1.38E+4	7.55E+3	7.70E+3	3.14E+3	**1.15E+3**	**4.36E+3**	**2.78E+3**	**2.78E+3**	**8.02E+2**
23	1.13E+3	3.09E+3	1.91E+3	1.84E+3	5.73E+2	**3.63E+2**	**3.74E+2**	**3.68E+2**	**3.69E+2**	**2.81E+0**
24	**2.21E+2**	1.15E+3	6.11E+2	6.20E+2	2.43E+2	4.41E+2	**4.70E+2**	**4.52E+2**	**4.53E+2**	**8.55E+0**
25	**2.37E+2**	5.89E+2	3.37E+2	3.90E+2	1.25E+2	3.12E+2	**3.70E+2**	**3.32E+2**	**3.35E+2**	**1.70E+1**
26	**1.16E+2**	8.31E+2	4.22E+2	5.00E+2	2.75E+2	2.11E+2	**2.31E+2**	**2.17E+2**	**2.20E+2**	**6.86E+0**
27	4.09E+3	5.02E+3	4.75E+3	4.73E+3	2.66E+2	**3.28E+3**	**3.96E+3**	**3.48E+3**	**3.54E+3**	**2.10E+2**
28	1.20E+4	1.93E+4	**1.42E+4**	1.49E+4	2.63E+3	**1.10E+4**	**1.64E+4**	1.44E+4	**1.37E+4**	**2.02E+3**
29	5.68E+8	2.15E+9	1.32E+9	1.34E+9	4.50E+8	**2.99E+5**	**2.33E+6**	**8.89E+5**	**1.09E+6**	**6.38E+5**
30	1.08E+7	9.37E+7	3.56E+7	4.07E+7	2.35E+7	**1.33E+5**	**5.40E+5**	**2.98E+5**	**2.80E+5**	**1.28E+5**

search capability than DA in the exploration and exploitation phases of optimization problems of different types and dimensions.

Finally, the performance of the proposed MIDA was compared with the performance of some improved dragonfly algorithms in the literature. These DA versions are Adaptive Resistance and Stamina Strategy-based Dragonfly Algorithm (ARSSDA) [39], Elite Opposition learning and Exponential function adaptive steps based Dragonfly Algorithm (EOEDA) [38], Gaussian Mutational Bare-bone Dragonfly Algorithm (GGBDA) [37], and Quantum-behaved and Gaussian mutational Dragonfly Algorithm (QGDA) [36]. In Table 9, the MIDA and the other DA versions are compared for CEC2014 benchmarks with 30D. The table gives the mean and standard deviation values of the optimization results with 51 runs of MIDA and other DA versions. The ranking results of all algorithms are given at the end of each benchmark, and the average ranking results of the MIDA and DA versions for all benchmarks are in the last row of the table. The results of the DA versions in the comparison table are taken from the literature. For QGDA,

Table 6
Results of the Wilcoxon rank-sum test with 5% significance between MIDA and DA for CEC2014 benchmarks

	D					D			
No	**10**	**30**	**50**	**100**	**No**	**10**	**30**	**50**	**100**
1	**8.44E-18**	**3.30E-18**	**8.15E-5**	**8.15E-5**	16	**2.99E-13**	**1.04E-13**	2.12E-1	**3.86E-3**
2	**1.58E-12**	**3.30E-18**	**8.15E-5**	**8.15E-5**	17	**1.34E-15**	**4.70E-18**	**8.15E-5**	**8.15E-5**
3	**7.48E-17**	**3.30E-18**	**8.15E-5**	**8.15E-5**	18	**2.32E-12**	**4.43E-18**	**8.15E-5**	**8.15E-5**
4	**6.49E-11**	**3.50E-18**	**8.15E-5**	**8.15E-5**	19	**3.30E-18**	**4.70E-18**	**8.15E-5**	**8.15E-5**
5	5.60E-2	**2.75E-5**	7.43E-1	1.00E+0	20	**1.01E-17**	**3.30E-18**	**8.15E-5**	**8.15E-5**
6	**2.58E-9**	**4.01E-17**	**2.03E-3**	**8.15E-5**	21	**1.74E-16**	**4.99E-18**	**8.15E-5**	**8.15E-5**
7	**6.23E-16**	**3.30E-18**	**8.15E-5**	**8.15E-5**	22	**3.04E-16**	**3.53E-15**	**1.07E-4**	**1.07E-4**
8	**3.30E-18**	**3.30E-18**	**8.15E-5**	**8.15E-5**	23	**2.73E-10**	**1.08E-15**	**8.15E-5**	**8.15E-5**
9	**2.21E-13**	**7.51E-18**	**1.62E-3**	**7.10E-3**	24	**2.01E-14**	**1.44E-8**	**8.15E-5**	**1.51E-2**
10	**1.13E-17**	**3.30E-18**	**8.15E-5**	**8.15E-5**	25	**2.94E-11**	**5.65E-14**	**8.15E-5**	8.44E-1
11	4.91E-1	6.88E-1	**1.81E-2**	7.62E-2	26	1.07E-1	**4.23E-16**	1.89E-1	**5.82E-3**
12	2.39E-1	8.46E-1	7.43E-1	8.96E-1	27	**6.63E-8**	**1.47E-14**	**8.15E-5**	**8.15E-5**
13	**3.43E-10**	**2.01E-17**	**8.15E-5**	**8.15E-5**	28	**6.87E-6**	**1.01E-8**	1.15E-1	3.93E-1
14	**7.21E-12**	**3.30E-18**	**8.15E-5**	**8.15E-5**	29	**1.39E-14**	**5.89E-16**	**1.26E-2**	**8.15E-5**
15	**1.07E-14**	**3.30E-18**	**8.15E-5**	**8.15E-5**	30	**6.01E-15**	**3.30E-18**	**8.15E-5**	**8.15E-5**

only the results of the F13-F19 problems were included in the [36] study, so results for the other problems were not given. According to the mean value metric results among DA versions, the proposed MIDA is in second rank. From the comparison results, the average ranking result of ARSSDA, which is in the first place, was 1.90 and that of MIDA was 2.23. These values indicate that MIDA performs close to that of ARSSDA.

3.9. Comparison results for the CEC2020 benchmark problems

In this subsection, the proposed MIDA and the original DA were evaluated for the CEC2020 benchmark suite, one of the most recently introduced datasets in the field of numerical optimization. This test suite consists of ten extremely difficult optimization problems. One of these problems is a unimodal function, three of them are multi-modal functions, three are hybrid functions, and the other three are composite functions. All optimization benchmarks are minimization problems [84]. In searching for the global solution points of these minimization problems, the algorithms are run 30 times for different dimensions (D=5, 10, 15, and 20). The search space for all problems is in the range of -100 to 100. In the search for the global solution point, reaching the maximum number of function evaluations was used as the stopping criterion of the algorithms. The maximum number of function evaluations is selected as 5e4 for all dimensions of the problems. The population size (the number of dragonflies) is determined as 100 for all dimensions. The evaluation results of MIDA and DA exclude F6 and F7 in the 5D case.

Table 10 summarizes the statistical results of MIDA and original DA for CEC2020 benchmarks with 5D, 10D, 15D, and 20D. The statistical metrics in this table are best, worst, median, mean, and standard deviation (std) metrics. In the table, better results are shown in bold in the evaluation of both algorithms. Looking at the results of the test suit with 5D, the proposed MIDA outperforms the original DA in all metrics except best metric. In the best metric results of the benchmarks with 5D, the MIDA is better than DA for all test problems except for F2 and F8. In the 10-dimensional CEC2020 results, among all benchmarks, MIDA outperforms DA by 100% for the best metric, 90% for the worst, mean, and median metrics, and 80% for the standard deviation metric. The 15D benchmark results of the table show the superiority of MIDA over DA in all benchmarks and all metrics, except for the worst and std metric results of the F2. Finally, from the 20-dimensional CEC 2020 benchmark results, it is understood that the proposed MIDA structure is 100% more successful than the original DA in all test problems for all metrics except the standard deviation metric. To summarize, in the CEC2020 results, MIDA achieved a success rate of 95% for 5D benchmarks, 90% for 10D benchmarks, and 96% for 15D and 20D benchmarks.

Table 7
Comparison Results I of MIDA and other Meta-heuristics for CEC2014 benchmarks of 30D.

		MFO	PSO	DA	SCA	WOA	CCS	MIDA
FN1	*Mean*	7.59E+7	8.19E+7	4.49E+8	2.24E+8	3.24E+7	2.93E+9	1.07E+7
	Std	9.77E+7	8.27E+9	3.88E+8	7.18E+7	1.42E+7	8.68E+8	6.70E+6
	Rank	3	4	6	5	2	7	1
FN2	*Mean*	1.36E+10	2.81E+4	3.67E+10	1.60E+10	5.05E+6	9.48E+10	4.93E+7
	Std	8.42E+9	5.30E+2	1.55E+10	3.12E+9	4.79E+6	6.53E+9	2.10E+7
	Rank	4	1	6	5	2	7	3
FN3	*Mean*	8.99E+4	2.09E+1	1.81E+5	3.79E+4	2.97E+4	1.01E+7	2.50E+3
	Std	4.98E+4	2.77E+1	9.61E+4	7.28E+3	2.10E+4	1.32E+7	2.63E+3
	Rank	5	1	6	4	3	7	2
FN4	*Mean*	1.14E+3	9.01E+3	4.86E+3	1.02E+3	1.76E+2	2.43E+4	1.71E+2
	Std	1.13E+3	1.85E+2	3.72E+3	2.04E+2	4.70E+1	3.94E+3	5.04E+1
	Rank	4	6	5	3	2	7	1
FN5	*Mean*	2.04E+1	2.19E+2	2.08E+1	2.09E+1	2.04E+1	5.03E+2	2.09E+1
	Std	1.75E-1	6.01E+3	1.17E-1	5.00E-2	1.80E-1	6.33E-2	7.07E-2
	Rank	1	5	2	3	1	7	3
FN6	*Mean*	2.40E+1	6.78E+3	3.72E+1	3.44E+1	3.48E+1	6.04E+2	2.46E+1
	Std	3.33E+0	2.46E+0	3.71E+0	2.61E+0	3.61E+0	2.00E+0	4.38E+0
	Rank	1	7	5	3	4	6	2
FN7	*Mean*	1.17E+2	1.61E+0	3.54E+2	1.33E+2	1.03E+0	1.66E+3	1.45E+0
	Std	6.91E+1	2.67E+1	1.59E+2	2.91E+1	5.00E-2	1.28E+2	1.44E-1
	Rank	4	3	6	5	1	7	2
FN8	*Mean*	1.43E+2	1.10E+3	2.89E+2	2.36E+2	1.76E+2	1.16E+3	2.07E+1
	Std	3.81E+1	1.24E+1	4.77E+2	2.08E+1	3.73E+1	2.51E+1	4.59E+0
	Rank	2	6	5	4	3	7	1
FN9	*Mean*	2.23E+2	2.30E+6	3.01E+2	2.67E+2	2.13E+2	1.33E+3	1.15E+2
	Std	6.06E+1	8.06E+7	6.73E+1	2.04E+1	5.22E+1	3.95E+1	3.14E+1
	Rank	3	7	5	4	2	6	1
FN10	*Mean*	3.47E+3	4.43E+1	6.57E+3	5.88E+3	3.77E+3	8.06E+3	1.28E+3
	Std	8.85E+2	7.82E+3	6.79E+2	4.51E+2	5.30E+2	3.28E+2	5.48E+2
	Rank	3	1	6	5	4	7	2
FN11	*Mean*	4.15E+3	6.50E+5	6.92E+3	7.04E+3	4.50E+3	8.83E+3	7.03E+3
	Std	6.90E+2	6.35E+2	7.41E+2	2.77E+2	7.58E+2	5.25E+2	5.53E+2
	Rank	1	7	3	5	2	6	4
FN12	*Mean*	4.30E-1	3.73E+2	2.21E+0	2.45E+0	1.67E+0	1.21E+3	2.19E+0
	Std	2.60E-1	2.76E+2	5.26E-1	2.30E-1	3.60E-1	3.95E-1	4.75E-1
	Rank	1	6	4	5	2	7	3
FN13	*Mean*	2.21E+0	2.14E+2	5.14E+0	2.89E+0	5.00E-1	1.31E+3	5.57E-1
	Std	1.34E+0	1.58E+2	1.65E+0	3.30E-1	1.20E-1	1.57E+0	1.15E-1
	Rank	3	6	5	4	1	7	2
FN14	*Mean*	3.54E+1	9.05E+2	1.31E+2	4.11E+1	2.80E-1	1.67E+3	3.34E-1
	Std	2.47E+1	1.92E+3	5.52E+1	5.51E+0	4.00E-2	4.59E+1	1.45E-1
	Rank	3	6	5	4	1	7	2
FN15	*Mean*	2.23E+5	1.04E+7	2.85E+5	2.82E+3	7.00E+1	1.81E+6	2.31E+1
	Std	5.77E+5	6.54E+4	4.25E+5	3.62E+3	2.52E+1	8.34E+5	4.90E+0
	Rank	4	7	5	3	2	6	1
FN16	*Mean*	1.27E+1	2.49E+7	1.32E+1	1.28E+1	1.26E+1	1.61E+3	1.23E+1
	Std	5.30E-1	2.21E+9	3.23E-1	3.10E-1	5.80E-1	2.60E-1	5.70E-1
	Rank	3	7	5	4	2	6	1

Table 7
Comparison Results I of MIDA and other Meta-heuristics for CEC2014 benchmarks of 30D (continued).

		MFO	PSO	DA	SCA	WOA	CCS	MIDA
FN17	*Mean*	3.39E+6	1.26E+4	1.61E+7	6.61E+6	4.33E+6	3.82E+8	5.35E+5
	Std	4.07E+6	3.30E+2	1.36E+7	2.94E+6	2.22E+6	1.73E+8	6.14E+5
	Rank	3	1	6	5	4	7	2
FN18	*Mean*	5.19E+6	4.80E-2	6.11E+8	1.85E+8	1.55E+4	1.21E+10	1.65E+4
	Std	3.61E+7	1.43E+0	8.31E+8	8.68E+7	4.80E+4	3.86E+9	1.83E+4
	Rank	4	1	6	5	2	7	3
FN19	*Mean*	7.36E+1	2.91E+1	2.68E+2	9.08E+1	4.48E+1	2.73E+3	2.12E+1
	Std	5.32E+1	1.48E+1	1.08E+2	2.42E+1	3.14E+1	3.70E+2	2.27E+1
	Rank	4	2	6	5	3	7	1
FN20	*Mean*	5.67E+4	1.34E+1	2.78E+5	1.31E+4	2.04E+4	5.99E+6	8.13E+2
	Std	4.34E+4	4.16E+2	5.77E+5	4.83E+3	1.06E+4	1.25E+7	7.12E+2
	Rank	5	1	6	3	4	7	2
FN21	*Mean*	7.83E+5	3.88E+2	8.23E+6	1.48E+6	9.48E+5	1.85E+8	8.45E+4
	Std	1.18E+6	2.70E-1	8.77E+6	8.55E+5	9.91E+5	9.29E+7	1.01E+5
	Rank	3	1	6	5	4	7	2
FN22	*Mean*	8.67E+4	7.40E-1	1.12E+3	7.54E+2	7.49E+2	3.43E+6	4.39E+2
	Std	2.29E+4	1.19E+1	4.04E+2	1.25E+2	1.99E+2	3.84E+6	1.97E+2
	Rank	6	1	5	4	3	7	2
FN23	*Mean*	3.71E+2	2.94E+3	5.76E+2	3.70E+2	3.31E+2	3.73E+3	3.16E+2
	Std	3.98E+1	3.10E-1	1.98E+2	1.37E+1	6.24E+0	4.91E+2	3.66E-1
	Rank	4	6	5	3	2	7	1
FN24	*Mean*	2.76E+2	1.05E+6	2.81E+2	2.01E+2	2.06E+2	2.53E+3	2.34E+2
	Std	2.73E+1	4.44E+7	4.14E+1	9.10E-1	5.95E+0	6.83E+0	4.51E+0
	Rank	4	7	5	1	2	6	3
FN25	*Mean*	2.14E+2	1.64E+1	2.40E+2	2.27E+2	2.25E+2	2.66E+3	2.10E+2
	Std	7.65E+0	1.52E+4	2.28E+1	6.18E+0	1.66E+1	5.95E+1	3.23E+0
	Rank	3	1	6	5	4	7	2
FN26	*Mean*	1.03E+2	2.83E+5	1.04E+2	1.02E+2	1.00E+2	2.79E+3	1.01E+2
	Std	1.50E+0	1.99E+2	1.69E+0	5.60E-1	1.00E-1	1.06E+2	1.35E-1
	Rank	4	7	5	3	1	6	2
FN27	*Mean*	9.21E+2	2.28E+1	1.09E+3	7.22E+2	9.05E+2	7.07E+3	4.93E+2
	Std	2.23E+2	2.15E+0	3.58E+2	2.91E+2	4.15E+2	8.01E+2	1.75E+2
	Rank	5	1	6	3	4	7	2
FN28	*Mean*	1.12E+3	2.59E+0	2.24E+3	2.00E+3	2.15E+3	1.41E+4	1.59E+3
	Std	1.57E+2	5.98E+1	5.47E+2	3.17E+2	4.80E+2	1.91E+3	4.08E+2
	Rank	2	1	6	4	5	7	3
FN29	*Mean*	3.06E+6	2.71E+2	2.15E+7	1.34E+7	4.38E+6	1.10E+9	1.64E+6
	Std	3.62E+6	5.04E+2	1.09E+7	7.63E+6	4.68E+6	2.92E+8	5.19E+6
	Rank	3	1	6	5	4	7	2
FN30	*Mean*	5.89E+4	1.18E+7	4.79E+5	2.48E+5	8.20E+4	2.35E+7	1.14E+4
	Std	5.40E+4	8.87E+4	3.77E+5	8.39E+4	6.61E+4	1.48E+7	1.01E+4
	Rank	2	6	5	4	3	7	1
Average rank		**3.23**	**3.90**	**5.27**	**4.03**	**2.63**	**6.77**	**1.97**
Overall rank		**3**	**4**	**6**	**5**	**2**	**7**	**1**

Table 8
Comparison Results II of MIDA and other Meta-heuristics for CEC2014 benchmarks of 30D.

		COA	EHO	ASMO	VNBA	RFO	HFA	MIDA
FN1	*Mean*	1.03E+8	4.83E+8	1.20E+8	1.21E+8	1.93E+8	3.34E+7	1.07E+7
	Std	1.30E+6	1.30E+8	3.55E+7	4.42E+7	4.94E+7	1.63E+5	6.70E+6
	Rank	3	7	4	5	6	2	1
FN2	*Mean*	2.06E+9	3.82E+10	1.95E+9	2.81E+8	1.35E+10	2.14E+9	4.93E+7
	Std	6.91E+5	7.67E+9	2.81E+8	1.24E+8	2.58E+9	6.91E+5	2.10E+7
	Rank	4	7	3	2	6	5	1
FN3	*Mean*	1.78E+5	5.64E+4	2.45E+5	2.63E+4	8.73E+4	1.66E+5	2.50E+3
	Std	2.09E+2	6.17E+3	7.65E+4	1.64E+4	6.27E+3	1.79E+2	2.63E+3
	Rank	6	3	7	2	4	5	1
FN4	*Mean*	3.98E+2	5.45E+3	7.11E+2	6.29E+2	2.18E+3	4.88E+2	1.71E+2
	Std	8.54E-1	1.17E+3	3.56E+1	7.41E+1	4.07E+2	2.52E-1	5.04E+1
	Rank	2	7	5	4	6	3	1
FN5	*Mean*	4.88E+2	5.05E+2	5.11E+2	5.15E+2	4.99E+2	4.98E+2	2.09E+1
	Std	3.62E-3	5.80E-2	9.73E-2	5.43E-2	4.59E-2	3.88E-4	7.07E-2
	Rank	2	5	6	7	4	3	1
FN6	*Mean*	4.37E+2	6.18E+2	6.22E+2	6.16E+2	6.00E+2	5.99E+2	2.46E+1
	Std	1.65E+0	1.81E+0	2.79E+0	2.24E+0	1.63E+0	1.36E-2	4.38E+0
	Rank	2	6	7	5	4	3	1
FN7	*Mean*	7.01E+2	9.89E+2	7.04E+2	7.02E+2	7.01E+2	7.00E+2	1.45E+0
	Std	1.01E+1	4.51E+1	3.15E+0	1.18E+0	1.60E+1	1.12E-2	1.44E-1
	Rank	3	6	5	4	3	2	1
FN8	*Mean*	4.21E+2	1.05E+3	1.06E+3	8.15E+2	8.02E+2	8.01E+2	2.07E+1
	Std	1.18E+0	1.93E+1	1.96E+1	3.41E+0	1.24E+1	6.82E-2	4.59E+0
	Rank	2	6	7	5	4	3	1
FN9	*Mean*	9.02E+2	1.21E+3	1.17E+3	1.03E+3	9.03E+2	9.02E+2	1.15E+2
	Std	2.28E+2	3.04E+1	1.84E+1	2.69E+1	2.09E+1	1.17E+2	3.14E+1
	Rank	2	6	5	4	3	2	1
FN10	*Mean*	1.87E+3	7.69E+3	9.01E+3	1.20E+3	3.18E+3	1.00E+3	1.28E+3
	Std	1.82E+3	3.53E+2	4.41E+2	7.82E+1	2.48E+2	1.37E+3	5.48E+2
	Rank	4	6	7	2	5	1	3
FN11	*Mean*	4.98E+3	7.96E+3	9.80E+3	4.96E+3	4.43E+3	1.10E+3	7.03E+3
	Std	5.21E+2	3.49E+2	4.16E+2	5.89E+2	3.85E+2	3.03E+0	5.53E+2
	Rank	4	6	7	3	2	1	5
FN12	*Mean*	1.21E+3	1.21E+3	1.21E+3	1.21E+3	1.21E+3	1.21E+3	2.19E+0
	Std	1.56E-1	3.10E-1	8.09E-1	1.37E-1	2.15E-1	2.16E-1	4.75E-1
	Rank	2	2	2	2	2	2	1
FN13	*Mean*	1.31E+3	1.31E+3	1.31E+3	1.31E+3	1.31E+3	1.31E+3	5.57E-1
	Std	2.42E-1	2.11E-1	1.97E-1	1.08E-1	2.80E-1	1.56E-1	1.15E-1
	Rank	2	2	2	2	2	2	1
FN14	*Mean*	1.41E+3	1.47E+3	1.41E+3	1.41E+3	1.41E+3	1.41E+3	3.34E-1
	Std	8.74E-1	1.04E+1	1.34E+0	1.63E-1	7.71E+0	2.36E-1	1.45E-1
	Rank	2	3	2	2	2	2	1
FN15	*Mean*	3.84E+4	2.89E+5	1.53E+3	1.52E+3	1.09E+5	3.40E+4	2.31E+1
	Std	2.28E+0	1.14E+5	2.18E+1	2.46E+1	3.85E+4	9.34E+0	4.90E+0
	Rank	5	7	3	2	6	4	1
FN16	*Mean*	1.61E+3	1.61E+3	1.61E+3	1.61E+3	1.61E+3	1.61E+3	1.23E+1
	Std	8.51E-3	2.44E-1	2.06E-1	5.08E-1	2.54E-1	2.06E-3	5.70E-1
	Rank	2	3	2	2	2	2	1

Table 8
Comparison Results II of MIDA and other Meta-heuristics for CEC2014 benchmarks of 30D(continued).

		COA	EHO	ASMO	VNBA	RFO	HFA	MIDA
FN17	*Mean*	6.41E+6	1.15E+7	8.77E+6	1.47E+7	9.00E+6	6.53E+6	5.35E+5
	Std	5.54E+5	4.10E+6	3.78E+6	6.02E+6	3.09E+6	2.51E+4	6.14E+5
	Rank	2	6	4	7	5	3	1
FN18	*Mean*	2.18E+8	3.83E+8	9.30E+7	1.24E+7	2.03E+8	2.02E+8	1.65E+4
	Std	8.27E+4	1.45E+8	2.91E+7	6.87E+6	4.90E+7	6.14E+3	1.83E+4
	Rank	6	7	3	2	5	4	1
FN19	*Mean*	2.54E+3	2.09E+3	1.90E+3	1.94E+3	1.91E+3	1.91E+3	2.12E+1
	Std	2.05E+1	4.75E+1	4.57E+0	4.05E+1	2.80E+1	1.36E-1	2.27E+1
	Rank	6	5	2	4	3	3	1
FN20	*Mean*	8.81E+4	2.28E+4	7.12E+5	3.70E+4	5.87E+4	9.87E+3	8.13E+2
	Std	5.85E+3	5.46E+3	6.42E+5	2.13E+4	9.43E+3	1.16E+3	7.12E+2
	Rank	6	3	7	4	5	2	1
FN21	*Mean*	2.51E+5	2.74E+6	4.34E+6	3.67E+6	2.51E+6	2.05E+5	8.45E+4
	Std	3.51E+6	1.22E+6	2.07E+6	1.77E+6	7.88E+5	8.37E+3	1.01E+5
	Rank	3	5	7	6	4	2	1
FN22	*Mean*	8.21E+4	3.27E+3	3.37E+3	2.91E+3	2.28E+4	5.60E+4	4.39E+2
	Std	4.58E+2	1.97E+2	1.73E+2	1.48E+2	1.50E+2	9.62E+0	1.97E+2
	Rank	7	3	4	2	5	6	1
FN23	*Mean*	2.63E+3	2.43E+3	2.60E+3	2.63E+3	2.32E+3	2.32E+3	3.16E+2
	Std	1.19E+1	1.82E-2	7.32E+0	2.88E+1	1.19E+1	1.20E-1	3.66E-1
	Rank	5	3	4	5	2	2	1
FN24	*Mean*	2.41E+3	2.52E+3	2.61E+3	2.60E+3	2.42E+3	2.42E+3	2.34E+2
	Std	3.21E-1	5.54E-3	1.85E+0	1.94E+1	3.75E+0	5.45E-1	4.51E+0
	Rank	2	4	6	5	3	3	1
FN25	*Mean*	5.29E+3	2.62E+3	2.67E+3	2.69E+3	2.52E+3	2.52E+3	2.10E+2
	Std	2.62E+0	1.90E-4	4.88E+0	5.08E+0	1.93E+0	2.65E+0	3.23E+0
	Rank	6	3	4	5	2	2	1
FN26	*Mean*	2.62E+3	2.63E+3	2.65E+3	2.67E+3	2.62E+3	2.62E+3	1.01E+2
	Std	1.13E+1	1.58E+0	1.58E-1	1.14E-1	1.13E+1	5.41E+0	1.35E-1
	Rank	2	3	4	5	2	2	1
FN27	*Mean*	2.72E+3	2.81E+3	4.17E+3	4.23E+3	2.72E+3	2.72E+3	4.93E+2
	Std	3.65E-1	3.37E-3	9.74E+1	9.60E+1	3.45E+1	4.99E-1	1.75E+2
	Rank	2	3	4	5	2	2	1
FN28	*Mean*	2.82E+3	2.91E+3	4.16E+3	4.46E+3	2.82E+3	2.82E+3	1.59E+3
	Std	1.06E+0	5.99E-3	1.17E+2	3.50E+2	8.14E+1	1.19E+0	4.08E+2
	Rank	2	3	4	5	2	2	1
FN29	*Mean*	1.26E+7	3.62E+5	1.81E+6	3.11E+6	6.79E+6	1.80E+7	1.64E+6
	Std	1.23E+2	2.60E+4	3.15E+6	4.11E+6	1.37E+4	1.23E+2	5.19E+6
	Rank	6	1	3	4	5	7	2
FN30	*Mean*	3.88E+5	5.41E+5	8.86E+4	4.92E+4	3.29E+5	3.91E+5	1.14E+4
	Std	2.57E+2	6.55E+5	3.12E+4	2.00E+4	2.22E+5	9.19E+2	1.01E+4
	Rank	5	7	3	2	4	6	1
Average rank		**3.57**	**4.60**	**4.43**	**3.80**	**3.67**	**2.93**	**1.23**
Overall rank		**3**	**7**	**6**	**5**	**4**	**2**	**1**

Table 9
Comparison Results of MIDA and improved DA versions for CEC2014 benchmarks of 30D.

		EOEDA	ARSSDA	DA	GGBDA	QGDA	MIDA
FN1	*Mean*	1.66E+1	0.00E+0	4.49E+8	3.34E+7	-	1.07E+7
	Std	1.67E+1	0.00E+0	3.88E+8	2.36E+7	-	6.70E+6
	Rank	2	1	5	4	-	3
FN2	*Mean*	5.85E+2	9.04E+0	3.67E+10	5.78E+7	-	4.93E+7
	Std	5.08E+1	1.02E+1	1.55E+10	1.32E+7	-	2.10E+7
	Rank	2	1	5	4	-	3
FN3	*Mean*	6.52E+1	0.00E+0	1.81E+5	2.25E+2	-	2.50E+3
	Std	3.45E+0	0.00E+0	9.61E+4	1.02E+3	-	2.63E+3
	Rank	2	1	5	3	-	4
FN4	*Mean*	1.11E+2	0.00E+0	4.86E+3	5.95E+2	-	1.71E+2
	Std	2.23E+1	0.00E+0	3.72E+3	8.76E+1	-	5.04E+1
	Rank	2	1	5	4	-	3
FN5	*Mean*	2.61E+1	2.00E+1	2.08E+1	5.21E+2	-	2.09E+1
	Std	2.47E-3	1.57E-3	1.17E-1	5.59E-2	-	7.07E-2
	Rank	4	1	2	5	-	3
FN6	*Mean*	2.12E+1	2.04E+1	3.72E+1	6.20E+2	-	2.46E+1
	Std	7.27E-2	5.97E-2	3.71E+0	4.02E+0	-	4.38E+0
	Rank	2	1	4	5	-	3
FN7	*Mean*	1.28E+3	7.18E+2	3.54E+2	7.02E+2	-	1.45E+0
	Std	8.27E+1	1.67E+1	1.59E+2	1.40E-1	-	1.44E-1
	Rank	5	4	2	3	-	1
FN8	*Mean*	1.09E+3	8.72E+2	2.89E+2	8.90E+2	-	2.07E+1
	Std	1.21E+1	9.31E+0	4.77E+2	1.48E+1	-	4.59E+0
	Rank	5	3	2	4	-	1
FN9	*Mean*	1.22E+3	9.74E+2	3.01E+2	1.07E+3	-	1.15E+2
	Std	7.64E+0	1.77E+1	6.73E+1	3.54E+1	-	3.14E+1
	Rank	5	3	2	4	-	1
FN10	*Mean*	7.63E+1	3.03E+1	6.57E+3	2.16E+3	-	1.28E+3
	Std	4.24E+0	1.55E+0	6.79E+2	4.06E+2	-	5.48E+2
	Rank	2	1	5	4	-	3
FN11	*Mean*	8.28E+1	3.95E+1	6.92E+3	4.31E+2	-	7.03E+3
	Std	2.09E+0	3.62E+0	7.41E+2	5.81E+2	-	5.53E+2
	Rank	2	1	4	3	-	5
FN12	*Mean*	1.20E+2	9.20E-2	2.21E+0	1.20E+3	-	2.19E+0
	Std	2.78E-2	9.92E-2	5.26E-1	6.47E-1	-	4.75E-1
	Rank	4	1	3	5	-	2
FN13	*Mean*	1.31E+3	1.30E+3	5.14E+0	1.30E+3	2.50E+3	5.57E-1
	Std	1.07E-1	6.07E-2	1.65E+0	1.10E-1	1.34E-2	1.15E-1
	Rank	4	3	2	3	5	1
FN14	*Mean*	1.64E+1	1.40E+1	1.31E+2	1.40E+3	2.60E+3	3.34E-1
	Std	2.18E-1	1.81E-2	5.52E+1	4.94E-2	9.64E-3	1.45E-1
	Rank	3	2	4	5	6	1
FN15	*Mean*	1.44E+4	2.17E+0	2.85E+5	1.52E+3	2.70E+3	2.31E+1
	Std	4.90E+3	2.56E-1	4.25E+5	3.87E+0	9.26E-5	4.90E+0
	Rank	5	1	6	3	4	2
FN16	*Mean*	1.61E+3	1.61E+3	1.32E+1	1.61E+3	2.90E+3	1.23E+1
	Std	4.43E-1	9.35E-1	3.23E-1	3.80E-1	6.60E-4	5.70E-1
	Rank	5	3	2	4	6	1

Table 9
Comparison Results of MIDA and improved DA versions for CEC2014 benchmarks of 30D(continued).

		EOEDA	ARSSDA	DA	GGBDA	QGDA	MIDA
FN17	*Mean*	7.44E+4	1.93E+2	1.61E+7	2.17E+6	3.00E+3	5.35E+5
	Std	1.22E+5	7.51E+1	1.36E+7	2.72E+6	1.84E-5	6.14E+5
	Rank	3	1	6	5	2	4
FN18	*Mean*	1.48E+5	1.59E+6	6.11E+8	1.52E+4	2.03E+4	1.65E+4
	Std	3.46E+5	3.18E+6	8.31E+8	5.23E+4	2.56E+4	1.83E+4
	Rank	4	5	6	1	3	2
FN19	*Mean*	1.96E+2	3.08E+0	2.68E+2	1.92E+3	3.45E+3	2.12E+1
	Std	4.92E+0	6.08E+0	1.08E+2	8.24E+0	1.93E+2	2.27E+1
	Rank	3	1	4	5	6	2
FN20	*Mean*	4.09E+1	1.31E+1	2.78E+5	2.28E+3	-	8.13E+2
	Std	3.77E+1	3.23E+0	5.77E+5	6.93E+1	-	7.12E+2
	Rank	2	1	5	4	-	3
FN21	*Mean*	1.14E+4	2.37E+3	8.23E+6	1.92E+5	-	8.45E+4
	Std	6.15E+3	2.56E+3	8.77E+6	2.87E+5	-	1.01E+5
	Rank	2	1	5	4	-	3
FN22	*Mean*	2.91E+3	2.14E+2	1.12E+3	2.67E+3	-	4.39E+2
	Std	2.03E+2	8.16E+1	4.04E+2	1.37E+2	-	1.97E+2
	Rank	5	1	3	4	-	2
FN23	*Mean*	2.66E+2	2.63E+2	5.76E+2	2.50E+3	-	3.16E+2
	Std	6.11E+0	0.00E+0	1.98E+2	8.22E-2	-	3.66E-1
	Rank	2	1	4	5	-	3
FN24	*Mean*	2.61E+3	2.60E+3	2.81E+2	2.60E+3	-	2.34E+2
	Std	1.44E+1	3.51E-4	4.14E+1	4.26E-2	-	4.51E+0
	Rank	4	3	2	3	-	1
FN25	*Mean*	2.71E+2	2.71E+2	2.40E+2	2.70E+3	-	2.10E+2
	Std	1.28E+0	2.29E-1	2.28E+1	1.40E-3	-	3.23E+0
	Rank	4	3	2	5	-	1
FN26	*Mean*	2.70E+2	1.07E+2	1.04E+2	2.70E+3	-	1.01E+2
	Std	3.09E-1	1.63E-2	1.69E+0	1.80E-1	-	1.35E-1
	Rank	4	3	2	5	-	1
FN27	*Mean*	3.42E+3	2.00E+2	1.09E+3	2.90E+3	-	4.93E+2
	Std	4.11E+2	0.00E+0	3.58E+2	1.89E-3	-	1.75E+2
	Rank	5	1	3	4	-	2
FN28	*Mean*	5.54E+3	3.72E+3	2.24E+3	3.00E+3	-	1.59E+3
	Std	3.79E+2	8.35E+1	5.47E+2	2.89E-2	-	4.08E+2
	Rank	5	4	2	3	-	1
FN29	*Mean*	5.63E+7	2.62E+2	2.15E+7	3.11E+3	-	1.64E+6
	Std	5.26E+7	0.00E+0	1.09E+7	5.43E+0	-	5.19E+6
	Rank	5	1	4	2	-	3
FN30	*Mean*	6.30E+4	4.05E+4	4.79E+5	3.59E+3	-	1.14E+4
	Std	7.68E+4	3.08E+4	3.77E+5	6.09E+2	-	1.01E+4
	Rank	4	3	5	1	-	2
Average rank		**3.53**	**1.90**	**3.70**	**3.80**	**4.57**	**2.23**
Overall rank		**3**	**1**	**4**	**5**	**6**	**2**

Table 10
Comparison Results of MIDA and original DA for CEC2020 benchmarks.

		DA					MIDA				
Dim	No	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
5	1	1.01E+0	4.47E+8	1.32E+4	4.95E+7	1.16E+8	**0.00E+0**	**1.27E+4**	**5.84E+1**	**1.35E+3**	**3.03E+3**
	2	**7.69E+1**	7.41E+2	3.99E+2	3.81E+2	1.95E+2	7.95E+1	**5.55E+2**	**2.92E+2**	**3.01E+2**	**1.31E+2**
	3	2.61E+0	3.88E+1	1.63E+1	1.62E+1	7.70E+0	**6.13E-1**	**2.34E+1**	**7.33E+0**	**7.71E+0**	**5.32E+0**
	4	1.07E-2	2.52E+1	1.13E+0	2.61E+0	4.91E+0	**0.00E+0**	**8.59E-1**	**3.51E-1**	**3.73E-1**	**2.12E-1**
	5	2.14E+2	3.44E+4	3.05E+3	6.65E+3	1.00E+4	**9.97E-1**	**4.15E+1**	**1.05E+1**	**1.46E+1**	**1.21E+1**
	6	-	-	-	-	-	-	-	-	-	-
	7	-	-	-	-	-	-	-	-	-	-
	8	**6.00E-8**	7.32E+1	2.15E+1	2.39E+1	1.40E+1	1.86E-1	**2.60E+1**	**1.59E+1**	**1.52E+1**	**5.55E+0**
	9	1.00E+2	3.50E+2	2.09E+2	2.29E+2	8.85E+1	**5.56E-3**	**2.05E+2**	**1.00E+2**	**1.07E+2**	**3.77E+1**
	10	3.00E+2	3.82E+2	3.52E+2	3.60E+2	1.83E+1	**3.00E+2**	**3.47E+2**	**3.47E+2**	**3.43E+2**	**1.34E+1**
10	1	4.82E+6	5.42E+9	5.98E+8	1.37E+9	1.62E+9	**3.18E+3**	**8.37E+6**	**3.95E+4**	**3.58E+5**	**1.52E+6**
	2	6.50E+2	**1.91E+3**	**1.34E+3**	**1.33E+3**	**3.29E+2**	**6.33E+2**	1.92E+3	1.43E+3	1.38E+3	3.42E+2
	3	1.97E+1	2.37E+2	7.49E+1	8.90E+1	5.01E+1	**1.73E+1**	**5.40E+1**	**3.32E+1**	**3.38E+1**	**9.60E+0**
	4	4.11E+0	8.88E+4	1.19E+1	3.59E+3	1.64E+4	**7.57E-1**	**3.53E+0**	**1.71E+0**	**1.65E+0**	**6.90E-1**
	5	7.44E+3	3.24E+6	5.97E+4	4.37E+5	8.61E+5	**3.20E+2**	**1.96E+4**	**5.11E+3**	**7.75E+3**	**6.91E+3**
	6	5.97E-1	8.33E+1	5.64E+0	1.47E+1	1.85E+1	**4.80E-1**	**1.82E+1**	**1.24E+0**	**3.27E+0**	**5.43E+0**
	7	9.48E+2	3.31E+5	4.07E+3	2.13E+4	5.95E+4	**5.39E+1**	**3.87E+3**	**7.51E+2**	**8.90E+2**	**8.08E+2**
	8	4.71E+1	5.18E+2	1.70E+2	2.14E+2	1.22E+2	**3.87E+1**	**4.32E+2**	**1.10E+2**	**1.18E+2**	**6.27E+1**
	9	1.04E+2	5.76E+2	3.91E+2	3.92E+2	**6.71E+1**	**1.01E+2**	**3.64E+2**	**3.48E+2**	**2.93E+2**	9.66E+1
	10	4.06E+2	1.02E+3	5.44E+2	6.17E+2	1.79E+2	**3.98E+2**	**5.28E+2**	**4.25E+2**	**4.27E+2**	**3.11E+1**
15	1	5.40E+7	1.89E+10	1.64E+9	4.37E+9	5.37E+9	**2.44E+6**	**4.35E+7**	**8.55E+6**	**1.06E+7**	**7.75E+6**
	2	1.76E+3	**3.56E+3**	2.73E+3	2.65E+3	**4.40E+2**	**1.08E+3**	3.60E+3	**2.51E+3**	**2.44E+3**	5.93E+2
	3	1.00E+2	4.29E+2	1.91E+2	2.07E+2	9.16E+1	**3.14E+1**	**8.02E+1**	**6.05E+1**	**5.91E+1**	**1.25E+1**
	4	3.80E+1	1.18E+6	5.22E+2	4.70E+4	2.15E+5	**1.60E+0**	**9.47E+0**	**4.29E+0**	**4.54E+0**	**2.01E+0**
	5	1.72E+5	1.15E+7	3.67E+6	3.83E+6	2.14E+6	**5.36E+2**	**9.80E+5**	**2.66E+4**	**1.04E+5**	**2.06E+5**
	6	1.99E+1	8.72E+2	2.13E+2	3.22E+2	3.26E+2	**1.04E+1**	**8.42E+1**	**1.59E+1**	**3.27E+1**	**3.18E+1**
	7	4.75E+4	2.13E+6	5.42E+5	6.69E+5	5.36E+5	**1.64E+1**	**1.49E+5**	**4.72E+2**	**5.99E+3**	**2.71E+4**
	8	1.84E+2	3.36E+3	8.35E+2	1.34E+3	1.08E+3	**1.14E+2**	**2.50E+3**	**1.17E+2**	**5.09E+2**	**8.10E+2**
	9	3.99E+2	6.55E+2	4.93E+2	5.09E+2	5.13E+1	**3.94E+2**	**5.21E+2**	**4.11E+2**	**4.16E+2**	**2.33E+1**
	10	5.69E+2	2.75E+3	1.03E+3	1.09E+3	4.79E+2	**4.22E+2**	**6.65E+2**	**6.10E+2**	**5.81E+2**	**7.12E+1**
20	1	2.21E+8	2.56E+10	5.80E+9	7.28E+9	5.64E+9	**5.72E+6**	**3.76E+7**	**1.56E+7**	**1.66E+7**	**7.30E+6**
	2	1.92E+3	5.44E+3	4.12E+3	3.77E+3	8.68E+2	**1.63E+3**	**4.50E+3**	**3.45E+3**	**3.43E+3**	**7.56E+2**
	3	1.67E+2	5.78E+2	3.37E+2	3.69E+2	1.23E+2	**5.43E+1**	**1.42E+2**	**8.81E+1**	**8.94E+1**	**1.73E+1**
	4	8.74E+1	1.05E+5	1.30E+4	2.08E+4	2.61E+4	**4.77E+0**	**1.46E+1**	**8.42E+0**	**8.59E+0**	**2.79E+0**
	5	2.20E+5	3.85E+7	5.39E+6	7.40E+6	7.87E+6	**5.83E+3**	**1.21E+6**	**1.46E+5**	**2.32E+5**	**3.02E+5**
	6	4.01E+2	4.49E+2	4.36E+2	4.30E+2	**2.07E+1**	**3.01E+0**	**1.44E+2**	**2.73E+1**	**5.19E+1**	5.73E+1
	7	1.51E+5	7.07E+6	1.06E+6	1.49E+6	1.55E+6	**2.57E+3**	**2.09E+5**	**2.36E+4**	**4.32E+4**	**4.69E+4**
	8	2.50E+2	5.42E+3	4.13E+3	3.74E+3	**1.55E+3**	**1.25E+2**	**5.15E+3**	**1.50E+2**	**2.30E+3**	2.36E+3
	9	5.00E+2	8.31E+2	6.11E+2	6.29E+2	7.34E+1	**4.30E+2**	**5.42E+2**	**4.79E+2**	**4.84E+2**	**2.44E+1**
	10	5.14E+2	3.42E+3	1.07E+3	1.38E+3	9.05E+2	**4.13E+2**	**5.46E+2**	**4.70E+2**	**4.69E+2**	**4.18E+1**

Table 11
Computational complexity of MIDA and DA

D	T0	T1	DA		MIDA	
			$\widehat{T2}$	($\widehat{T2}$ -T1)/T0	$\widehat{T2}$	($\widehat{T2}$ -T1)/T0
5	9.37E-2	8.10E-1	9.27E+1	9.81E+2	6.39E+1	6.73E+2
10	9.37E-2	1.79E+0	9.40E+1	9.84E+2	6.77E+1	7.04E+2
15	9.37E-2	3.71E+0	9.73E+1	9.98E+2	7.43E+1	7.53E+2

The computational complexity of the proposed MIDA and original DA is given in Table 11. The computational complexity of both algorithms was calculated on a PC with Intel(R) Core(TM) i7-6500U CPU@2.50 GHz with 8 GB RAM. First, T0 is determined, which is the execution time of the following program.

```
for i=1:1000000 do
    x = 0.55 + (double) i;
    x=x+x; x=x/2; x=x*x; x=sqrt(x);
    x=log(x); x=exp(x); x=x/(x+2);
end for
```

Second, the execution times (T1) are calculated in three different problem dimensions (D=5, 10, and 15) of the F1 benchmark for 2000000 evaluations. Then, the computation times (T2) of the algorithms for 2000000 evaluations are found using the MIDA and DA structures of F1 in the same problem dimensions. This step is performed five times to obtain the T2 values and calculate the average $\widehat{T2}$. The $T0$, $T1$, $\widehat{T2}$, and $(\widehat{T2} - T1)/T0$ values for the computational complexity of both algorithms are summarized in Table 11. Here, the computational complexities of MIDA and DA are presented according to the problem dimension. The $\widehat{T2}$ results in this table show that the computation times of the proposed MIDA are less than those of the original DA for all problem dimensions. It is also understood that the three mechanisms added to the original dragonfly algorithm do not increase the execution time of MIDA and even decrease the computation time for different problem sizes.

3.10. MIDA-ANFIS hybrid model for wind speed forecasting problem

The proposed MIDA is adapted to the ANFIS model for short-term wind speed forecasting as a real-world application in this section. The proposed MIDA-ANFIS hybrid model is presented in Fig. 16. Four sequential wind speed dataset inputs ($x(t), x(t-1), x(t-2), x(t-3)$) are used in the ANFIS model based on MIDA for one-step short-term wind speed forecasting.ANFIS has two parameters to be updated by the training process: premise and consequent parameters. The parameters of the Gauss membership function and the defuzzification layer are represented by the premise and consequent parameters, respectively. In the optimization process, the problem dimension is one of the significant criteria. The proposed MIDA algorithm is used in the optimization of ANFIS parameters. To evaluate the model performance, the results are compared with those of the standard DA based ANFIS model.

The hybrid models are applied to monthly wind speed data collected in different seasons in Turkey (Fig. 17).Following normalization, 70% of the data is employed for training the model, with the remaining 30% used for testing the model's performance. All implemented models are run separately 50 times to eliminate the errors caused by the randomness of the model parameters. The predicted values deviation from the target value during the test and training phases shows model performance. For this aim, the root mean square error (RMSE), mean square error (MSE), and mean absolute percentage error (MAPE) mean absolute error (MAE) metrics are used.

The performance metrics can be expressed by (18) through (21) as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \tilde{y}_i)^2}{N}} \tag{18}$$

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

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

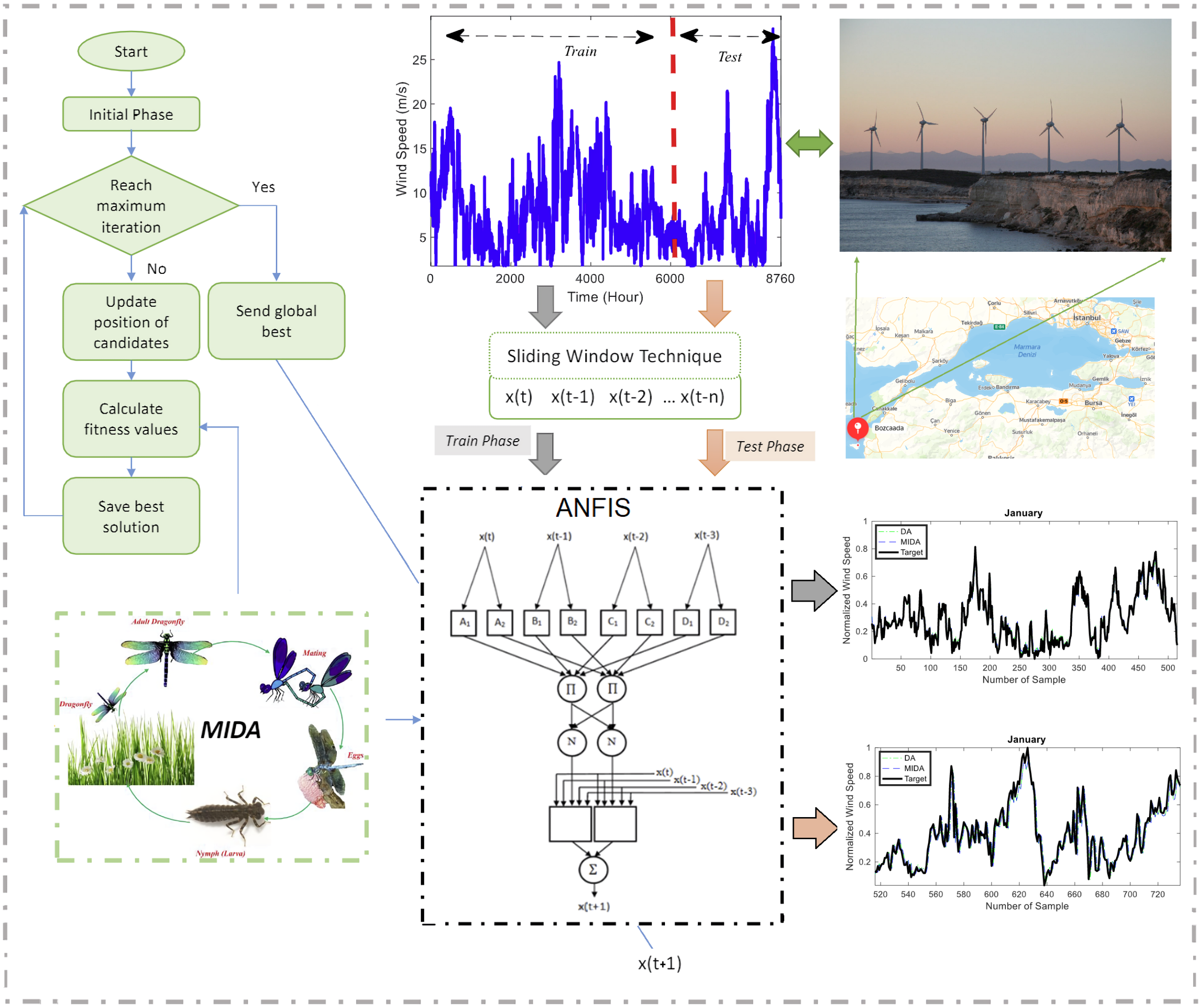


Figure 16: The structure of the ANFIS and MIDA hybrid forecasting model.

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

Where y_i and $\tilde{y}_i$ are the observed and predicted wind speed values, respectively, and the total number of data is indicated by N. Figures 18 and 19 show the comparison of the prediction results from the implemented models for the training and test phases, respectively. It is shown that the ANFIS model with optimized by the MIDA display outperformance for training and test phase of the short-term forecasting.Furthermore, Fig. 20 depicts the scatter plots of MIDA-based ANFIS model performance between observed and forecasted values to show the degree of correlation.

Fig. 21 illustrates the distribution of error metrics for the ANFIS model error metrics for DA and MIDA. These boxplots are the optimization results obtained with 50 runs for the ANFIS test phase. It can be concluded that the MIDA is found to be more effective and reliable than the DA for all months. The variation of performance metrics for each iteration for a sample month such as July is presented in Fig. 22. As can be seen, the MIDA has the lowest RMSE, MSE, MAE, and MAPE metrics values. Moreover, the statistical results of these changes for all months are calculated. Particularly, the MAE, RMSE, MSE, and MAPE can effectively reflect the differences between the original values and their corresponding forecasted values. Tables 12, 13, 14, and 15 provide a comprehensive comparison of the forecasting models used. It can be clearly seen that the MIDA-based ANFIS forecasting models outperform the corresponding DA-based ANFIS forecasting models in terms of all performance metrics on the four datasets with

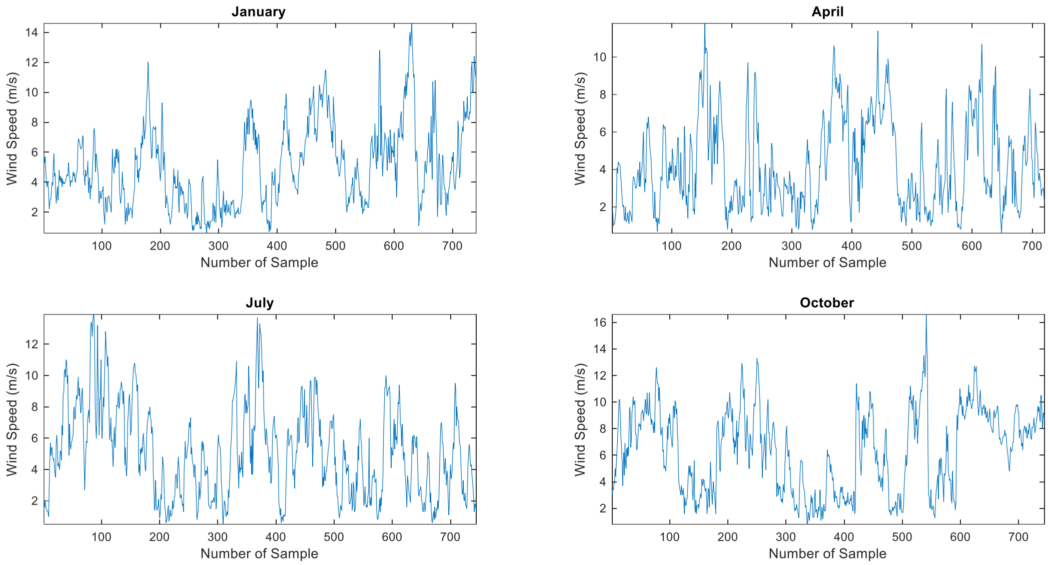


Figure 17: Original wind speed time series for seasonal analysis

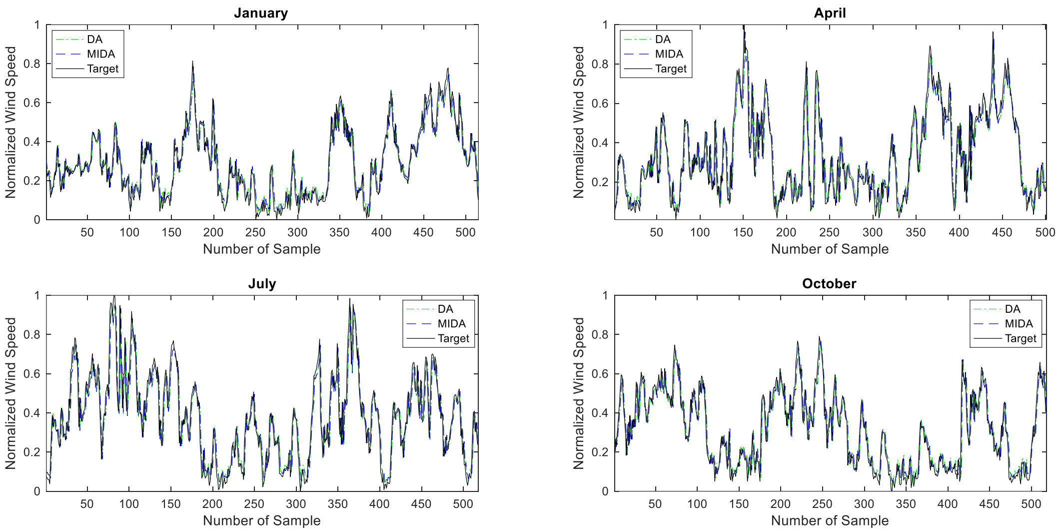


Figure 18: Training results for the best ANFIS parameters as obtained by DA and MIDA in 50 runs.

different seasons. The MSE mean values for test results of the MIDA hybrid model on the four-month datasets are found to be 0.902%, 0.994%, 0.534%, and 0.446%, which are lower than the corresponding results of the DA hybrid model. From the reported values, it can be concluded that the mutation-based improvement can enhance information and improve the accuracy of prediction. As such, the MIDA based ANFIS model can achieve enhanced short-term wind speed forecasting. It can also be applied in other applications related to forecasting, such as renewable power production and electrical load consumption, by taking the application specific aspects into account.

Finally, a comparison of six different models frequently used in time series forecasting with the proposed ANFIS-MIDA hybrid model was performed for short-term wind speed forecasting. These models are Back Propagation Neural Network (BPNN), Elman Neural Network, Long Short-Term Memory (LSTM), ANFIS, ANFIS-Genetic Algorithm hybrid model (ANFIS-GA), ANFIS-Particle Swarm Optimizer hybrid model (ANFIS-PSO). In BPNN model, the

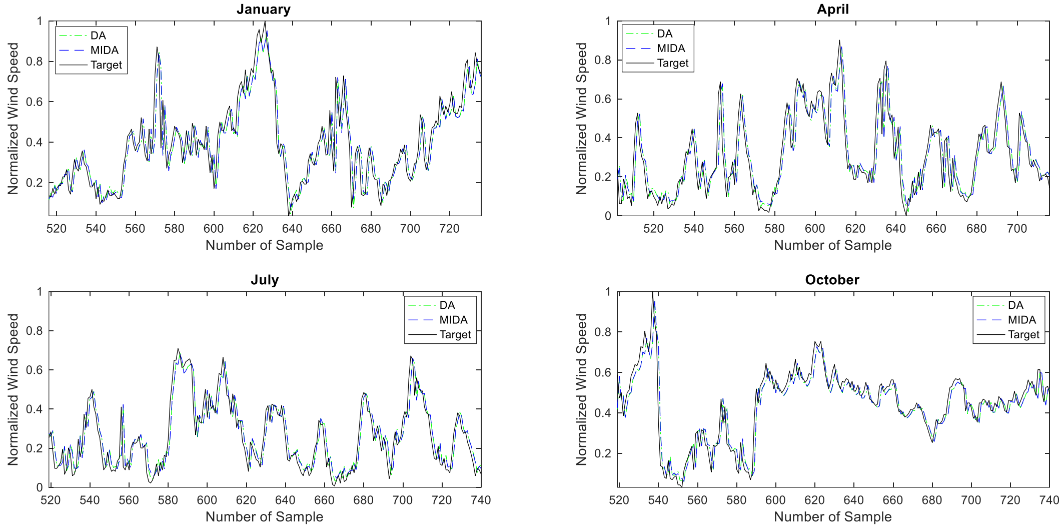


Figure 19: Test results for the best ANFIS parameters as obtained by DA and MIDA in 50 runs.

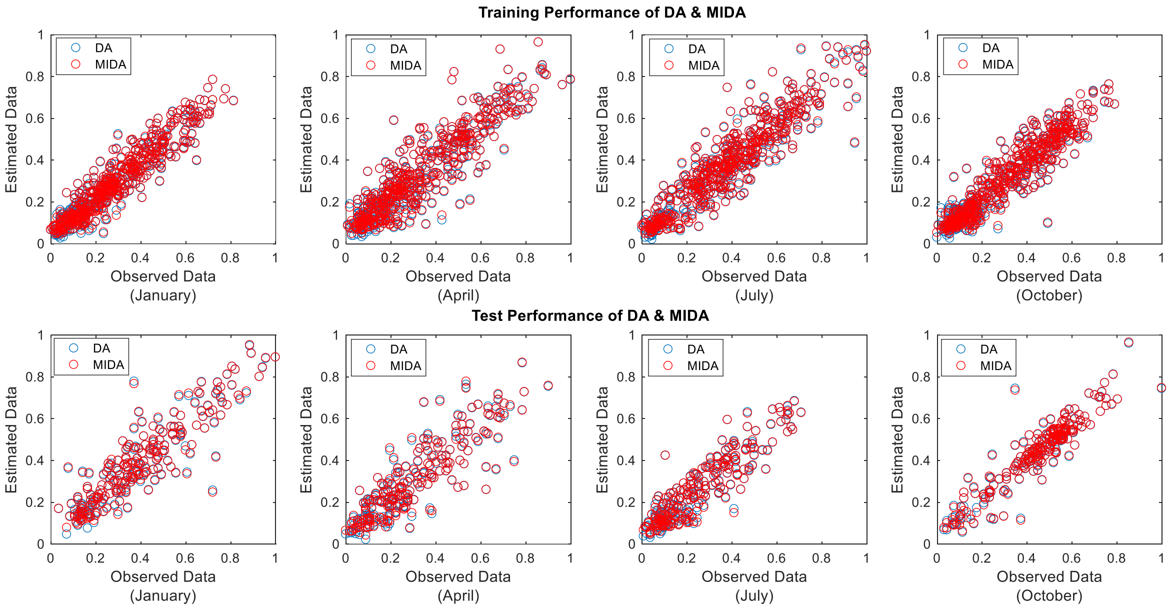


Figure 20: Scatter results for the best ANFIS parameters as obtained by DA and MIDA in 50 runs.

learning rate is 0.01, there are 3 hidden layers, each with a Tansig function, and an output layer with a Purelin function, Input nodes are 4 as in ANFIS-MIDA. Elman NN model consists of an input, a hidden, a context, and an output layer. The context layer uses the hidden layer's previous output values as input to itself. Here, the number of hidden layer neurons is 10, in the context layer, a 2-step delay is used in the input, Tansig is selected as transfer function in hidden layer, output layer transfer function is Purelin, and the learning rate is 0.01. In LSTM model, one hidden layer with five neurons is used, and the learning rate is taken as 0.01. In the simple ANFIS model, ANFIS-GA, and ANFIS-PSO models, only the training part differs from the ANFIS-MIDA model. Grid Partitioning is used for training in the simple ANFIS model, Genetic algorithm is used for training in the ANFIS-GA model, and Particle Swarm Optimizer is used for training the ANFIS-PSO model. In ANFIS-GA model, GA parameters are determined as follows: the crossover

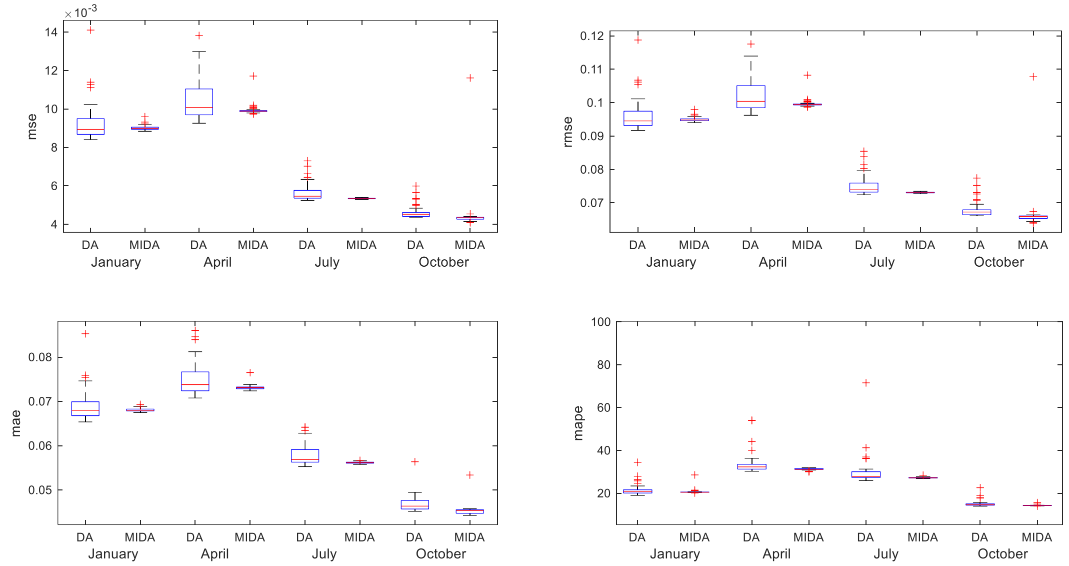


Figure 21: Distribution of ANFIS model error metrics in test phase using DA and MIDA

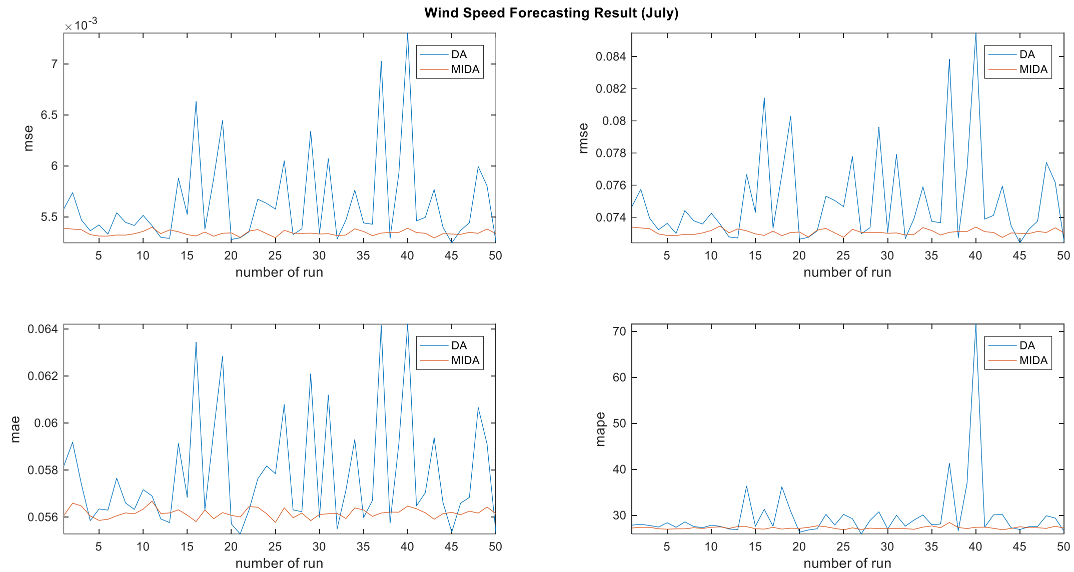


Figure 22: The short term wind speed forecasting error metric results for July.

probability is 0.4, the mutation probability is 0.7, population size is 20, and maximum iteration is 1000. The PSO parameters in the ANFIS-PSO hybrid model are as follows: personal learning coefficient (c_1) is 1, global learning coefficient (c_2) is 2, population size is 20, and maximum iteration is 1000.

Table 16 summarizes the comparison results among ANFIS-MIDA hybrid model and other models for short-term wind speed forecasting. Here, all model results are presented on an annual basis. For this process, wind speed data for each season (January, April, July, and October) were first separated as 70% for training and 30% for testing, and then a single training/test data was prepared by combining four different seasonal data. This table includes MSE, RMSE, MAE, and MAPE metric results of all models. At the same time, the results are presented in two parts as training and testing phases. The performance rankings of the models are given below each error metric result. The bottom two rows

Table 12
MSE metric results of ANFIS-DA and ANFIS-MIDA with 50 independent runs

		Training Results (MSE)				Test Results (MSE)			
		January	April	July	October	January	April	July	October
ANFIS DA	best	3.70E-3	8.15E-3	7.22E-3	4.52E-3	**8.41E-3**	**9.26E-3**	**5.24E-3**	4.37E-3
	worst	5.43E-3	1.11E-2	9.72E-3	6.56E-3	1.41E-2	1.38E-2	7.30E-3	**6.00E-3**
	mean	3.93E-3	8.67E-3	7.68E-3	4.82E-3	9.29E-3	1.04E-2	5.64E-3	4.63E-3
	std	2.74E-4	6.34E-4	6.14E-4	3.72E-4	9.92E-4	1.03E-3	4.47E-4	**3.47E-4**
ANFIS MIDA	best	**3.67E-3**	**8.00E-3**	**7.08E-3**	**4.44E-3**	8.84E-3	9.73E-3	5.29E-3	**4.08E-3**
	worst	**3.74E-3**	**8.21E-3**	**7.24E-3**	**4.62E-3**	**9.60E-3**	**1.17E-2**	**5.40E-3**	1.16E-2
	mean	**3.72E-3**	**8.15E-3**	**7.19E-3**	**4.56E-3**	**9.02E-3**	**9.94E-3**	**5.34E-3**	**4.46E-3**
	std	**1.41E-5**	**3.58E-5**	**4.66E-5**	**3.80E-5**	**1.26E-4**	**2.69E-4**	**2.51E-5**	1.04E-3

Table 13
RMSE metric results of ANFIS-DA and ANFIS-MIDA with 50 independent runs

		Training Results (RMSE)				Test Results (RMSE)			
		January	April	July	October	January	April	July	October
ANFIS DA	best	6.08E-2	9.03E-2	8.50E-2	6.73E-2	**9.17E-2**	**9.62E-2**	**7.24E-2**	6.61E-2
	worst	7.37E-2	1.05E-1	9.86E-2	8.10E-2	1.19E-1	1.18E-1	8.55E-2	**7.74E-2**
	mean	6.26E-2	9.30E-2	8.76E-2	6.93E-2	9.63E-2	1.02E-1	7.50E-2	6.80E-2
	std	2.08E-3	3.29E-3	3.38E-3	2.56E-3	4.85E-3	4.93E-3	2.87E-3	**2.45E-3**
ANFIS MIDA	best	**6.06E-2**	**8.95E-2**	**8.42E-2**	**6.66E-2**	9.40E-2	9.87E-2	7.28E-2	**6.39E-2**
	worst	**6.12E-2**	**9.06E-2**	**8.51E-2**	**6.80E-2**	**9.80E-2**	**1.08E-1**	**7.35E-2**	1.08E-1
	mean	**6.10E-2**	**9.03E-2**	**8.48E-2**	**6.75E-2**	**9.50E-2**	**9.97E-2**	**7.31E-2**	**6.65E-2**
	std	**1.16E-4**	**1.98E-4**	**2.75E-4**	**2.82E-4**	**6.58E-4**	**1.30E-3**	**1.72E-4**	5.99E-3

Table 14
MAE metric results of ANFIS-DA and ANFIS-MIDA with 50 independent runs

		Training Results (MAE)				Test Results (MAE)			
		January	April	July	October	January	April	July	October
ANFIS DA	best	**4.60E-2**	6.64E-2	**6.29E-2**	5.04E-2	**6.54E-2**	**7.08E-2**	**5.53E-2**	4.52E-2
	worst	5.61E-2	7.69E-2	7.39E-2	6.13E-2	8.53E-2	8.60E-2	6.42E-2	5.64E-2
	mean	4.78E-2	6.87E-2	6.55E-2	5.23E-2	6.89E-2	7.51E-2	5.79E-2	4.68E-2
	std	1.53E-3	2.83E-3	2.56E-3	1.92E-3	3.40E-3	3.73E-3	2.38E-3	1.81E-3
ANFIS MIDA	best	4.62E-2	**6.59E-2**	6.30E-2	**4.97E-2**	6.75E-2	7.24E-2	5.58E-2	**4.43E-2**
	worst	**4.67E-2**	**6.68E-2**	**6.40E-2**	**5.12E-2**	**6.93E-2**	**7.65E-2**	**5.67E-2**	**5.34E-2**
	mean	**4.65E-2**	**6.65E-2**	**6.36E-2**	**5.07E-2**	**6.81E-2**	**7.32E-2**	**5.62E-2**	**4.53E-2**
	std	**1.29E-4**	**1.71E-4**	**2.69E-4**	**2.90E-4**	**3.67E-4**	**5.69E-4**	**1.99E-4**	**1.24E-3**

of the table show the average ranking and overall ranking results. Although the error metric results seem to be close to each other, the overall ranking clearly show that the best model is the ANFIS-MIDA model.

4. Discussion

DA is a new meta-heuristic algorithm presented to the literature and has been used by many researchers in different studies. Although the search capability of the algorithm seems promising, it has some weak points that can be improved. In this study, it is aimed to increase the search performance with improvements to overcome these weaknesses of DA. Here is a summary of the results with an overview of them:

Table 15
MAPE metric results of ANFIS-DA and ANFIS-MIDA with 50 independent runs

		Training Results (MAPE)				Test Results (MAPE)			
		January	April	July	October	January	April	July	October
ANFIS DA	best	21.38	**26.63**	**22.21**	21.80	**19.03**	30.30	**25.94**	14.54
	worst	78.75	61.85	34.21	81.79	34.44	54.07	71.56	18.69
	mean	24.05	29.67	23.78	25.93	21.43	33.57	29.83	15.58
	std	8.26	4.96	2.14	10.50	2.60	4.82	6.71	0.79
ANFIS MIDA	best	**21.10**	26.98	22.22	**21.43**	20.13	**30.02**	26.84	**14.02**
	worst	**22.37**	**27.92**	**22.69**	**23.00**	**28.57**	**31.92**	**28.42**	**15.63**
	mean	**21.85**	**27.46**	**22.42**	**22.49**	**20.71**	**31.23**	**27.26**	**14.33**
	std	**0.25**	**0.19**	**0.10**	**0.32**	**1.15**	**0.42**	**0.27**	**0.23**

Table 16
Comparison results of the ANFIS-MIDA and other models

		BPNN	Elman	LSTM	ANFIS	ANFIS-GA	ANFIS-PSO	ANFIS-MIDA
Training	*MSE*	5.97E-3	2.14E-2	6.01E-3	6.60E-3	6.08E-3	5.92E-3	5.89E-3
	Rank	3	7	4	6	5	2	1
	RMSE	7.71E-2	1.46E-1	7.75E-2	8.13E-2	7.80E-2	7.69E-2	7.67E-2
	Rank	3	7	4	6	5	2	1
	MAE	5.66E-2	1.14E-1	5.70E-2	6.05E-2	5.72E-2	5.70E-2	5.68E-2
	Rank	1	7	3	6	5	4	2
	MAPE	2.31E+1	4.10E+1	3.67E+1	2.51E+1	2.60E+1	2.36E+1	2.35E+1
	Rank	1	7	6	4	5	3	2
Test	*MSE*	7.22E-3	2.31E-2	7.34E-3	8.00E-3	7.33E-3	7.13E-3	7.16E-3
	Rank	3	7	5	6	4	1	2
	RMSE	8.50E-2	1.52E-1	8.57E-2	8.97E-2	8.56E-2	8.44E-2	8.46E-2
	Rank	3	7	5	6	4	1	2
	MAE	6.07E-2	1.18E-1	6.12E-2	6.50E-2	6.08E-2	6.06E-2	6.06E-2
	Rank	2	6	4	5	3	1	1
	MAPE	2.31E+1	3.89E+1	2.68E+1	2.50E+1	2.56E+1	2.33E+1	2.33E+1
	Rank	1	7	6	4	5	2	3
Average rank		2.13	6.88	4.63	5.38	4.50	2.00	1.75
Overall rank		3	7	5	6	4	2	1

- MIDA was created in this study by integrating mutation operator, boundary control, and greedy selection mechanisms into the original DA structure. The mutation operator is effective in finding the global optimum without getting stuck with the local optimum points. In the boundary control mechanism, the dragonflies updated in the standard DA are provided to take a random position in case of going outside the limit values of the search space, maximum 25% from the limit value. In the greedy selection mechanism, the search continues in the next iteration with the more fit dragonflies from the comparison of the fitness values of the previous dragonflies and the updated dragonflies.

- First, for the search performance analysis of the proposed MIDA, four of the CEC2014 test problems were taken and some basic meta-heuristic algorithm analyzes were performed. These analyzes are search history, convergence behavior, the average distance of the first dragonfly, the trajectory of the elite dragonfly, computational complexity, diversity behavior, and balance analysis. All analysis results were compared with the original DA. The analysis results show that the three different mechanisms in the proposed MIDA structure increase the search performance of the original DA, provide fast convergence, and have a more dominant exploitative behavior.

- Secondly, to evaluate the performance of the proposed MIDA, 30 minimization problems from CEC2014 test suite, and 10 minimization problems from CEC2020 test suite are utilized for different dimensions. Experimental results for all dimensions of CEC2014 benchmarks show that MIDA's performance is 91.33% better than the original DA. Again, statistical results obtained for all dimensions of CEC2020 benchmarks say that MIDA is 94.25% more successful than the original DA. The performance of the proposed MIDA was compared not only with the original DA, but also with twelve meta-heuristics and four different enhanced DA versions from the literature. MIDA ranked first in comparison with other meta-heuristics for the 30-dimensional CEC2014 test problems. As a result of the comparison with the DA versions, it took the second place after ARSSDA.

- Finally, the proposed MIDA is integrated into a real optimization problem. Optimization of the parameters of the ANFIS model, which is used for short-term wind speed estimation, is provided by the MIDA structure. Training and test results with one-year seasonal wind speed data were evaluated according to different error metrics (MSE, RMSE, MAE, and MAPE) for MIDA and original DA. In addition, the ANFIS-MIDA hybrid model was compared with six models used in time series estimation. Error metric results show that the performance of ANFIS-MIDA is in the first place.
- The proposed MIDA structure has a higher search performance compared to other meta-heuristic algorithms for CEC2014 and CEC2020 benchmarks. However, MIDA also has some limitations. If the solution of the optimization problem is close to the boundary of the search space, the convergence rate of the dragonflies decreases due to the boundary value control mechanism of MIDA. At the same time, the average distance of dragonflies to each other cannot reach exactly zero. Again, the greedy selection mechanism in the structure of MIDA contributes to the very rapid convergence of dragonflies in the search space, while reducing diversity compared to the original DA. While the mutation operator in MIDA provides the solution to avoid local optimal points, it may not always guarantee this. Therefore, more work needs to be done to address these limitations of MIDA.
- Potential future work on MIDA could be: further increasing the search performance by reducing the limits of MIDA, adapting it to multi-objective optimization problems, developing hybrid algorithms with different methodologies, applying it to different engineering problems.

5. Conclusion and Future Works

In this paper, the DA has been improved by incorporating three mechanisms, namely, mutation operation, boundary control, and greedy selection, to improve its performance in accuracy and exploration capability in the search space. The mechanisms have achieved the following objectives: The mutation operator avoids getting stuck at the local optimum point. The boundary control mechanism presents the new positions for the dragonflies that exceed the boundaries of the search space. Finally, the greedy selection mechanism, which is newly added to the DA, ensures the search for better dragonflies after each iteration. The improved model has been used to optimize ANFIS parameters. The performance of the improved based ANFIS model has been applied to short-term wind speed forecasting as a real-world problem.

The performance of the proposed MIDA over the original DA has been evaluated and shown its superiority in several analyses such as convergence, search history, trajectory, average distance, computational complexity, diversity, and balance. For validation, the MIDA has been tested on the 10, 30, 50, and 100 dimensions of the CEC2014 benchmarks, and 5, 10, 15, and 20 dimensions of CEC2020 benchmarks. The results confirmed that the proposed MIDA outperforms the original DA by 89.33% in 10-dimensional benchmarks, 91.33% in 30-dimensional benchmarks, 91.33% in 50-dimensional benchmarks, and 93.33% in 100-dimensional benchmarks according to all statistical metrics considered. Wilcoxon rank-sum test results prove the statistical significance of the proposed MIDA in comparison with DA. It was also shown that the MIDA displayed superiority over the some meta-heuristic algorithms (MFO, PSO, DA, SCA, WOA, CCS, COA, EHO, ASMO, VNBA, RFO, and HFA) for the 30-dimensional CEC2014 benchmark functions. The comparison results are ranked according to the mean metric of the 51 runs of the algorithms, and the performance of the proposed MIDA is clearly superior to other algorithms and ranks first. In addition, different DA versions (EOEDA, ARSSDA, GGBDA, and QGDA) from the literature were compared with the proposed MIDA for 30D CEC2014 benchmarks. In the results obtained here, MIDA ranks second in performance, but has a close ranking score with ARSSDA in the first place. Comparing the proposed MIDA with the original DA for the CEC2020 test problems, MIDA has 95% better results for 5-dimensional problems, 90% for 10-dimensional problems, 96% for 15-dimensional problems, and 96% for 20-dimensional problems for all statistical metrics. The computational complexity for F1 from the CEC2020 benchmarks was examined and it was seen that the computation time of MIDA was less than the original DA for different problem sizes.

The benchmark results confirmed that the proposed MIDA could be a good alternative to solve real-world optimization problems. Finally, as a real-world application, it was adapted for training ANFIS model parameters for use in short-term wind speed forecasting. Both training and testing results confirmed the better forecasting performance of the MIDA based ANFIS model over that of the DA based ANFIS counterpart.In addition, the ANFIS-MIDA hybrid model and six different models (BPNN, Elman NN, LSTM, ANFIS, ANFIS-GA, and ANFIS-PSO), which are frequently used in time series estimation, were compared. According to the error metric results obtained during the training and test of the models, the ANFIS-MIDA hybrid model is the best one compared to the other models.

In future studies on MIDA, the search performance of the algorithm can be increased further by introducing different solutions to the limitations of MIDA, which were mentioned in the previous section. The proposed MIDA is a good alternative to other meta-heuristic algorithms and therefore its application results can be examined in different engineering problems. The binary version of MIDA can be used for feature reduction in classification problems. In this study, single-objective optimization problems are discussed, and in future studies, multi-objective MIDA can be developed for multi-objective optimization problems.

6. Acknowledgement

This work was supported in part by the Scientific and Technological Research Council of Turkey through the International PostDoctoral Fellowship Program under Grant No: 1059B192001283.

References

[1] K. E. Adetunji, I. Hofsajer, A. M. Abu-Mahfouz, L. Cheng, A review of metaheuristic techniques for optimal integration of electrical units in distribution networks, IEEE Access (2020).

[2] I. Fister Jr, X.-S. Yang, I. Fister, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization, arXiv preprint arXiv:1307.4186 (2013).

[3] W. Sun, M. Tang, L. Zhang, Z. Huo, L. Shu, A survey of using swarm intelligence algorithms in iot, Sensors 20 (2020) 1420.

[4] Q.-T. Bui, Q.-H. Nguyen, X. L. Nguyen, V. D. Pham, H. D. Nguyen, V.-M. Pham, Verification of novel integrations of swarm intelligence algorithms into deep learning neural network for flood susceptibility mapping, Journal of Hydrology 581 (2020) 124379.

[5] D. Wei, Z. Wang, L. Si, C. Tan, Preaching-inspired swarm intelligence algorithm and its applications, Knowledge-Based Systems 211 (2021) 106552.

[6] M. Rostami, K. Berahmand, E. Nasiri, S. Forouzande, Review of swarm intelligence-based feature selection methods, Engineering Applications of Artificial Intelligence 100 (2021) 104210.

[7] M. Janga Reddy, D. Nagesh Kumar, Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review, H2Open Journal 3 (2021) 135–188.

[8] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, volume 4, IEEE, 1995, pp. 1942–1948.

[9] M. Dorigo, Optimization, learning and natural algorithms, Ph. D. Thesis, Politecnico di Milano (1992).

[10] X.-l. Li, An optimizing method based on autonomous animats: fish-swarm algorithm, Systems Engineering-Theory & Practice 22 (2002) 32–38.

[11] K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE control systems magazine 22 (2002) 52–67.

[12] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report, Technical report-tr06, Erciyes university, engineering faculty, computer …, 2005.

[13] X.-S. Yang, Firefly algorithms for multimodal optimization, in: International symposium on stochastic algorithms, Springer, 2009, pp. 169–178.

[14] S. Mirjalili, The ant lion optimizer, Advances in engineering software 83 (2015) 80–98.

[15] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Advances in Engineering Software 114 (2017) 163–191.

[16] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future generation computer systems 97 (2019) 849–872.

[17] F. S. Gharehchopogh, B. Abdollahzadeh, An efficient harris hawk optimization algorithm for solving the travelling salesman problem, Cluster Computing (2021) 1–25.

[18] S. Kaur, L. K. Awasthi, A. Sangal, G. Dhiman, Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, Engineering Applications of Artificial Intelligence 90 (2020) 103541.

[19] F. S. Gharehchopogh, An improved tunicate swarm algorithm with best-random mutation strategy for global optimization problems, Journal of Bionic Engineering (2022) 1–26.

[20] M. S. Kiran, Tsa: Tree-seed algorithm for continuous optimization, Expert Systems with Applications 42 (2015) 6686–6698.

[21] F. S. Gharehchopogh, Advances in tree seed algorithm: A comprehensive survey, Archives of Computational Methods in Engineering (2022) 1–24.

[22] G. Dhiman, V. Kumar, Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications, Advances in Engineering Software 114 (2017) 48–70.
[23] S. Ghafori, F. S. Gharehchopogh, Advances in spotted hyena optimizer: a comprehensive survey, Archives of computational methods in engineering (2021) 1–22.
[24] B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, Computers & Industrial Engineering 158 (2021) 107408.
[25] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, International Journal of Intelligent Systems 36 (2021) 5887–5958.
[26] H. Shayanfar, F. S. Gharehchopogh, Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems, Applied Soft Computing 71 (2018) 728–746.
[27] F. S. Gharehchopogh, B. Farnad, A. Alizadeh, A modified farmland fertility algorithm for solving constrained engineering problems, Concurrency and Computation: Practice and Experience 33 (2021) e6310.
[28] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Computing and Applications 27 (2016) 1053–1073.
[29] M. Hariharan, R. Sindhu, V. Vijean, H. Yazid, T. Nadarajaw, S. Yaacob, K. Polat, Improved binary dragonfly optimization algorithm and wavelet packet based non-linear features for infant cry classification, Computer Methods and Programs in Biomedicine 155 (2018) 39–51.
[30] B. Zhang, L. Xu, J. Zhang, Balancing and sequencing problem of mixed-model u-shaped robotic assembly line: Mathematical model and dragonfly algorithm based approach, Applied soft computing 98 (2021) 106739.
[31] B. Vedik, R. Kumar, R. Deshmukh, S. Verma, C. K. Shiva, Renewable energy-based load frequency stabilization of interconnected power systems using quasi-oppositional dragonfly algorithm, Journal of Control, Automation and Electrical Systems 32 (2021) 227–243.
[32] A. I. Hammouri, M. Mafarja, M. A. Al-Betar, M. A. Awadallah, I. Abu-Doush, An improved dragonfly algorithm for feature selection, Knowledge-Based Systems 203 (2020) 106131.
[33] M. Alshinwan, L. Abualigah, M. Shehab, M. Abd Elaziz, A. M. Khasawneh, H. Alabool, H. Al Hamad, Dragonfly algorithm: a comprehensive survey of its results, variants, and applications, Multimedia Tools and Applications (2021) 1–38.
[34] Y. Meraihi, A. Ramdane-Cherif, D. Acheli, M. Mahseur, Dragonfly algorithm: a comprehensive review and applications, Neural Computing and Applications (2020) 1–22.
[35] B. A. S. Emambocus, M. B. Jasser, A. Mustapha, A. Amphawan, Dragonfly algorithm and its hybrids: A survey on performance, objectives and applications, Sensors 21 (2021) 7542.
[36] C. Yu, Z. Cai, X. Ye, M. Wang, X. Zhao, G. Liang, H. Chen, C. Li, Quantum-like mutation-induced dragonfly-inspired optimization approach, Mathematics and Computers in Simulation 178 (2020) 259–289.
[37] L. Yuan, F. Kuang, S. Zhang, H. Chen, The gaussian mutational barebone dragonfly algorithm: From design to analysis, Symmetry 14 (2022) 331.
[38] J. Song, S. Li, Elite opposition learning and exponential function steps-based dragonfly algorithm for global optimization, in: 2017 IEEE International Conference on Information and Automation (ICIA), IEEE, 2017, pp. 1178–1183.
[39] Y. Yuan, S. Wang, L. Lv, X. Song, An adaptive resistance and stamina strategy-based dragonfly algorithm for solving engineering optimization problems, Engineering Computations (2020).
[40] G. I. Sayed, A. Tharwat, A. E. Hassanien, Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection, Applied Intelligence 49 (2019) 188–205.
[41] Ç. İ. Acı, H. Gülcan, A modified dragonfly optimization algorithm for single-and multiobjective problems using brownian motion, Computational Intelligence and Neuroscience 2019 (2019).
[42] X. Bao, H. Jia, C. Lang, Dragonfly algorithm with opposition-based learning for multilevel thresholding color image segmentation, Symmetry 11 (2019) 716.
[43] R. K. Sambandam, S. Jayaraman, Self-adaptive dragonfly based optimal thresholding for multilevel segmentation of digital images, Journal of King Saud University-Computer and Information Sciences 30 (2018) 449–461.
[44] R. Salgotra, U. Singh, S. Singh, G. Singh, S. Saha, A new set of mutation operators for dragonfly algorithm, Arabian Journal for Science and Engineering 46 (2021) 8761–8802.
[45] M. Abedi, F. S. Gharehchopogh, An improved opposition based learning firefly algorithm with dragonfly algorithm for solving continuous optimization problems, Intelligent Data Analysis 24 (2020) 309–338.
[46] M. Duan, H. Yang, B. Yang, X. Wu, H. Liang, Hybridizing dragonfly algorithm with differential evolution for global optimization, IEICE Transactions on Information and Systems 102 (2019) 1891–1901.
[47] Z. Han, J. Zhang, S. Lin, C. Liu, Research on the improved dragonfly algorithm-based flexible flow-shop scheduling, in: Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019), Springer, 2020, pp. 205–214.
[48] W. A. Ghanem, A. Jantan, A cognitively inspired hybridization of artificial bee colony and dragonfly algorithms for training multi-layer perceptrons, Cognitive Computation 10 (2018) 1096–1134.
[49] C. Shilaja, T. Arunprasath, Internet of medical things-load optimization of power flow based on hybrid enhanced grey wolf optimization and dragonfly algorithm, Future Generation Computer Systems 98 (2019) 319–330.
[50] M. A. Tawhid, K. B. Dsouza, Hybrid binary dragonfly enhanced particle swarm optimization algorithm for solving feature selection problems, Mathematical Foundations of Computing 1 (2018) 181.
[51] V. Veeramsetty, C. Venkaiah, D. Kumar, Hybrid genetic dragonfly algorithm based optimal power flow for computing lmp at dg buses for reliability improvement, Energy systems 9 (2018) 709–757.
[52] M. Abdel-Basset, Q. Luo, F. Miao, Y. Zhou, Solving 0–1 knapsack problems by binary dragonfly algorithm, in: International conference on intelligent computing, Springer, 2017, pp. 491–502.

[53] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, S. Mirjalili, Binary dragonfly algorithm for feature selection, in: 2017 International conference on new trends in computing sciences (ICTCS), IEEE, 2017, pp. 12–17.
[54] Y. Chen, Z. Wang, Wavelength selection for nir spectroscopy based on the binary dragonfly algorithm, Molecules 24 (2019) 421.
[55] H. Benimam, C. S. Moussa, M. Hentabli, S. Hanini, M. Laidi, Dragonfly-support vector machine for regression modeling of the activity coefficient at infinite dilution of solutes in imidazolium ionic liquids using σ-profile descriptors, Journal of Chemical & Engineering Data 65 (2020) 3161–3172.
[56] Y. Mesellem, A. A. El Hadj, M. Laidi, S. Hanini, M. Hentabli, Computational intelligence techniques for modeling of dynamic adsorption of organic pollutants on activated carbon, Neural Computing and Applications (2021) 1–20.
[57] L. Penghui, A. A. Ewees, B. H. Beyaztas, C. Qi, S. Q. Salih, N. Al-Ansari, S. K. Bhagat, Z. M. Yaseen, V. P. Singh, Metaheuristic optimization algorithms hybridized with artificial intelligence model for soil temperature prediction: Novel model, IEEE Access 8 (2020) 51884–51904.
[58] H. Tao, A. A. Ewees, A. O. Al-Sulttani, U. Beyaztas, M. M. Hameed, S. Q. Salih, A. M. Armanuos, N. Al-Ansari, C. Voyant, S. Shahid, et al., Global solar radiation prediction over north dakota using air temperature: development of novel hybrid intelligence model, Energy Reports 7 (2021) 136–157.
[59] S. V. Razavi-Termeh, K. Khosravi, A. Sadeghi-Niaraki, S.-M. Choi, V. P. Singh, Improving groundwater potential mapping using metaheuristic approaches, Hydrological Sciences Journal 65 (2020) 2729–2749.
[60] E. Dokur, U. Yüzgeç, M. Kurban, Performance comparison of hybrid neuro-fuzzy models using meta-heuristic algorithms for short-term wind speed forecasting, Electrica 21 (2021) 305–321.
[61] A. Jaafari, S. V. R. Termeh, D. T. Bui, Genetic and firefly metaheuristic algorithms for an optimized neuro-fuzzy prediction modeling of wildfire probability, Journal of environmental management 243 (2019) 358–369.
[62] L. M. Halabi, S. Mekhilef, M. Hossain, Performance evaluation of hybrid adaptive neuro-fuzzy inference system models for predicting monthly global solar radiation, Applied energy 213 (2018) 247–261.
[63] M. Deveci, N. Erdogan, U. Cali, J. Stekli, S. Zhong, Type-2 neutrosophic number based multi-attributive border approximation area comparison (mabac) approach for offshore wind farm site selection in usa, Engineering Applications of Artificial Intelligence 103 (2021) 104311.
[64] E. Dokur, N. Erdogan, M. E. Salari, C. Karakuzu, J. Murphy, Offshore wind speed short-term forecasting based on a hybrid method: Swarm decomposition and meta-extreme learning machine, Energy 248 (2022) 123595.
[65] Z. Shang, Z. He, Y. Chen, Y. Chen, M. Xu, Short-term wind speed forecasting system based on multivariate time series and multi-objective optimization, Energy 238 (2022) 122024.
[66] W. Sun, B. Tan, Q. Wang, Multi-step wind speed forecasting based on secondary decomposition algorithm and optimized back propagation neural network, Applied Soft Computing 113 (2021) 107894.
[67] L.-L. Li, Z.-F. Liu, M.-L. Tseng, K. Jantarakolica, M. K. Lim, Using enhanced crow search algorithm optimization-extreme learning machine model to forecast short-term wind power, Expert Systems with Applications 184 (2021) 115579.
[68] A. Altan, S. Karasu, E. Zio, A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer, Applied Soft Computing 100 (2021) 106996.
[69] G. Osório, J. Matias, J. Catalão, Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information, Renewable Energy 75 (2015) 301–307.
[70] L. Zhang, J. Wang, X. Niu, Wind speed prediction system based on data pre-processing strategy and multi-objective dragonfly optimization algorithm, Sustainable Energy Technologies and Assessments 47 (2021) 101346.
[71] J. S. R. Jang, Anfis: adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man, and Cybernetics 23 (1993) 665–685.
[72] H. Kilic, U. Yuzgec, C. Karakuzu, Improved antlion optimizer algorithm and its performance on neuro fuzzy inference system, Neural Network World 29 (2019) 235–254.
[73] J. J. Liang, B. Y. Qu, P. N. Suganthan, Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 635 (2013) 490.
[74] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowledge-based systems 89 (2015) 228–249.
[75] S. Mirjalili, Sca: a sine cosine algorithm for solving optimization problems, Knowledge-based systems 96 (2016) 120–133.
[76] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in engineering software 95 (2016) 51–67.
[77] G.-G. Wang, S. Deb, A. H. Gandomi, Z. Zhang, A. H. Alavi, Chaotic cuckoo search, Soft Computing 20 (2016) 3349–3362.
[78] J. Pierezan, L. D. S. Coelho, Coyote optimization algorithm: a new metaheuristic for global optimization problems, in: 2018 IEEE congress on evolutionary computation (CEC), IEEE, 2018, pp. 1–8.
[79] M. A. Elhosseini, R. A. El Sehiemy, Y. I. Rashwan, X. Gao, On the performance improvement of elephant herding optimization algorithm, Knowledge-Based Systems 166 (2019) 58–70.
[80] A. Sharma, A. Sharma, B. K. Panigrahi, D. Kiran, R. Kumar, Ageist spider monkey optimization algorithm, Swarm and Evolutionary Computation 28 (2016) 58–77.
[81] G.-G. Wang, M. Lu, X.-J. Zhao, An improved bat algorithm with variable neighborhood search for global optimization, in: 2016 IEEE congress on evolutionary computation (CEC), IEEE, 2016, pp. 1773–1778.
[82] D. Połap, M. Woźniak, Red fox optimization algorithm, Expert Systems with Applications 166 (2021) 114107.
[83] E. F. Veysari, et al., A new optimization algorithm inspired by the quest for the evolution of human society: human felicity algorithm, Expert Systems with Applications 193 (2022) 116468.
[84] C. Yue, K. Price, P. N. Suganthan, J. Liang, M. Z. Ali, B. Qu, N. H. Awad, P. P. Biswas, Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization, Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep 201911 (2019).