RESEARCH ARTICLE

Multi-strategy Enhanced Grey Wolf Optimizer for Numerical Optimization and Its Application to Feature Selection

Wenmin Xu [1] **and Wen Long** [1,*]

[1] School of Mathematics and Statistics, Guizhou University of Finance and Economics, Guiyang 550025, China

Abstract

Grey wolf optimizer (GWO) is an effective meta-heuristic technique which has been widely utilized to solve numerical optimization as well as real-world applications. However, GWO has some shortcomings, i.e., low solution accuracy, slow convergence, and easy stagnation at local optima in solving complex problems. To tackle these shortcomings, an enhanced GWO called EGWO is developed in this study. This enhancement is achieved by embedding three novel strategies into the basic GWO to improve its performance. Firstly, a new transition mechanism is designed instead of the original strategy to obtain a good transition from the exploration to exploitation. Secondly, the cuckoo search algorithm is introduced for the decision layer individuals (α, β, and δ) to further improve the local search capability. Thirdly, an adaptive position search equation is proposed by using a dynamical parameter to generate new potential candidate position. The effectiveness of EGWO is verified on 25 classical benchmarks, three engineering problems, and 16 feature selection problems. The results show that EGWO performs better than the original GWO and other meta-heuristic methods in terms of solution accuracy and convergence speed.

Keywords: grey wolf optimizer, cuckoo search, numerical optimization, Engineering optimization, feature selection.

1 Introduction

During the past few decades, many meta-heuristic based methods have been suggested in the literature and widely used to solve various function optimization and real-world applications [1, 2]. In this article, we concentrated on the grey wolf optimizer (GWO), which proposed by Mirjalili et al. [3]. The main reason to study the GWO is its special search mechanism, which is the leadership hierarchy of wolf swarm. During the past few years, GWO has received widely concerned by researchers and has been widely utilized for solving many real-world applications such as community detection [3], parameter extraction of PV models [4], feature selection [5], Optimal reactive power dispatch [6], power scheduling problem [7], data clustering [8], engineering problem [9],wind speed forecasting [10], imaging segmentation [11, 12],

Submitted: 02 September 2025
Accepted: 28 October 2025
Published: 29 November 2025

Vol. 1, **No.** 1, 2025.
10.62762/JMIA.2025.522565

***Corresponding author:**
Wen Long
lw227@mail.gufe.edu.cn

Citation
Xu, W., & Long, W. (2025). Multi-strategy Enhanced Grey Wolf Optimizer for Numerical Optimization and Its Application to Feature Selection. *Journal of Mathematics and Interdisciplinary Applications*, 1(1), 51–71.

path planning of UAVs [13], diagnosis of diabetes diseases [14], economic load dispatch [15], unit commitment problem [16], parameter identification of fuel cell [17], and many others [18].

Although the GWO has been widely applied to many real-world applications arising in various areas, some studies [19] indicate that its search performance will be reduced when applying to solve complex problems. To improve the search performance of GWO, many GWO variants have been developed in the literature. In the GWO, the conversion factor plays a crucial role in transition from global search to the local search processes. Thus, one active research interest is to study the conversion factor of the GWO. Several nonlinear strategies of the parameter have been suggested to achieve a good trade-off between global search and local search [20–23]. Compared with the basic GWO, these variants have performed well on benchmark test problems. In addition, the position search equation of the GWO is utilized to produce the new candidate solution based on the information of the decision layer individuals (α, β, and δ wolves). This equation is good at local search, but poor at global search. So, the motivation of modifying position search equation is to enhance its global search capability. Several modified versions of the position search equation have been developed to improve the exploration capability of the GWO [23, 24]. These modifications have obtained the better performance on benchmark optimization tasks as well as application problems. Due to different algorithms have different search characteristics, a straightforward idea is to hybridize GWO with other algorithms. Therefore, various hybrid algorithms based on the GWO and other meta-heuristic methods such as cuckoo search (CS) [25], gravitational search algorithm (GSA) [26], particle swarm optimization (PSO) [27], naked mole-rat algorithm (NMRA) [28], crow search algorithm (CSA) [29], genetic algorithm (GA) [30], Dragonfly Algorithm (DA) [31], flower pollination algorithm (FPA) [32], whale optimization algorithm (WOA) [33], Shuffled Frog Leaping Algorithm (SFLA) [34], multi-verse optimizer (MVO) [35] have been proposed in the literature. The experimental results indicated that these hybrid algorithms show well performance on benchmark functions as well as real-world application problems in terms of solution accuracy and convergence speed. Moreover, several new operators are introduced to enhance the search performance of the basic GWO algorithm. For example, there are the opposition-based learning [36], refraction- based learning [37], Lévy flight [20], random walk [38], mutation operator [39], evolutionary operator [40], astrophysics operator [41], and many others [42–44].

The above mentioned variants have tried to enhance the performance of the GWO by introducing the new mechanisms, there are still in some cases such as for complicated problems with high dimensionality and high nonlinearity, the algorithm suffers from the premature convergence and low accuracy. Moreover, the "No-free- lunch" theorem pointed that there is no any optimization algorithm which is used to solve all optimization problems. Motivation of these considerations, this study developed a novel GWO called multi-strategies enhanced grey wolf optimizer (EGWO). More specifically, the main contributions of our work are summarized:

- An enhanced grey wolf optimizer (EGWO) based on multi-strategies is developed for solving the function optimization and feature selection problems.
- A novel transition rule based on the nonlinear function is designed to achieve a better balance between exploration and exploitation.
- The cuckoo search algorithm is introduced for the current decision layer wolves to further accelerate the convergence.
- An adaptive position search equation by introducing a dynamic parameter is proposed to produce the promising candidate solution and improve the diversity of algorithm.
- We investigate the effectiveness and efficiency of EGWO using 25 classical benchmark functions, three engineering design, and 16 feature selection problems.

The rest of our study is structured as follows. Section 2 describes the basic GWO algorithm. Section 3 explains the implementation of three modified strategies. Section 4 tests the performance on different problems. Finally, Section 5 summarizes the conclusions of this study.

2 Grey wolf optimizer

GWO simulates the leadership hierarchy characteristics and the hunting behavior of grey wolves. In GWO, the individuals in the population are divided into four categories, i.e., α wolf is the best individual in the population, β wolf represents the second best individual, δ wolf denotes the third best individual, and ω wolves represent the other individuals in the population. The leadership hierarchy characteristics of grey wolves are given in Figure 1.

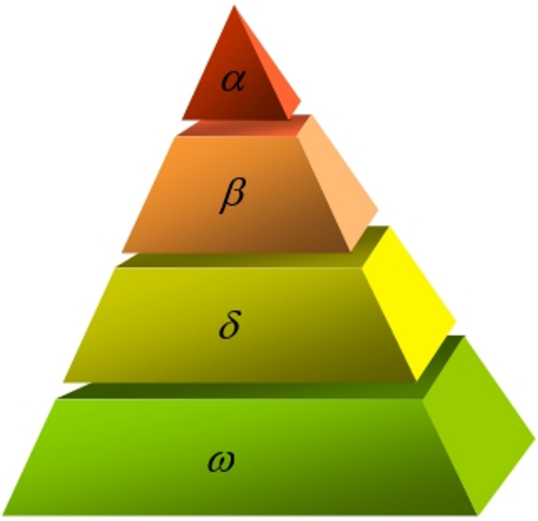

Figure 1. Leadership hierarchy characteristics of grey wolves.

In GWO, the following Eqs. (1) - (3) are introduced to model the encircling characteristic

$$X(t+1) = X_p(t) - A(t) \cdot |C(t) \cdot X_p(t) - X(t)| \quad (1)$$
$$A(t) = 2 \cdot a(t) \cdot r_1 - a(t) \quad (2)$$
$$C(t) = 2 \cdot r_2 \quad (3)$$

where X indicates the wolf position, X_α denotes the prey position, t indicates the iterative number, A and C are the coefficient factors, r_1 and r_2 are the random number. The a is also called transition parameter and is calculated by

$$a(t) = 2 - 2 \cdot \left(\frac{t}{T}\right) \quad (4)$$

where T indicates the maximum iterative number.

In the GWO, supposing that the decision layer individuals (i.e., α, β, and δ wolves) have enough capability for hunting the prey. Therefore, the other grey wolves in the population are attracted towards the decision layer individuals region and the mathematical model of the hunting characteristic is provided as follows.

$$Y_1 = X_\alpha(t) - A_\alpha(t) \cdot |C_\alpha(t) \cdot X_\alpha(t) - X(t)| \quad (5)$$
$$Y_2 = X_\beta(t) - A_\beta(t) \cdot |C_\beta(t) \cdot X_\beta(t) - X(t)| \quad (6)$$
$$Y_3 = X_\delta(t) - A_\delta(t) \cdot |C_\delta(t) \cdot X_\delta(t) - X(t)| \quad (7)$$
$$X(t+1) = (Y_1 + Y_2 + Y_3)/3 \quad (8)$$

where X_α, X_β and X_δ represent the position of the α, β, and δ wolves, the calculations of A_α, A_β, and A_δ are similar to A, the calculations of C_α, C_β, and C_δ are similar to C.

Algorithm 1 introduces the step-wise explanation of the original GWO.

```
Algorithm 1: Pseudo code of original GWO
Initialize the population of grey wolves
Initialize the parameters
Calculate the function values of each individual
X_α: the best individual
X_β: the second best individual
X_δ: the third best individual
t ← 0
while t < T do
    Update the position of each individual by
      using Eq. (8)
    Update the transition parameter a by using Eq.
      (4)
    Calculate the function values of each
      individual
    Update X_α, X_β, and X_δ
    t ← t + 1
end
Return X_α
```

3 Multi-strategies enhanced GWO algorithm

Despite the GWO is quite efficient for exploring the new unknown regions of the search space, however, it has some shortcomings namely low solution accuracy, unbalanced exploitation, and easy to trap into local optima when solving complex optimization problems. Therefore, the goal of this paper is to develop a novel variant for improving the search performance of the original GWO. Our study does not change the framework of the original GWO, and improves the performance of the GWO by embedding three strategies, namely the nonlinear transition parameter α, the cuckoo search algorithm and the adaptive position search equation.

3.1 Nonlinear transition parameter

The search capability of a meta-heuristic algorithm depends on how achieve a good balance between diversity and convergence over the optimization process. In the original GWO, the parameter a is called to the transition parameter, plays a crucial role while coordinating between diversity and convergence. However, in GWO, the parameter a is varied linearly over the iterative process of the algorithm, while many optimization problems require nonlinearly vary in the global and local search properties of an algorithm to escape from the locally optimal solutions. Thus, a potential research interest is to design the new transition parameter a rules in the GWO for boosting its search performance. In the literature, various modified strategies of the transition parameter a have been suggested [21–23]. In the original GWO, the larger values of $a(> 1)$ facilitates to the global search, while the smaller values of $a(< 1)$ is helpful for the local search of the solution space. Selecting an appropriate value of a is important to achieve a good balance between global search and local search of GWO. Due to the fact that the evolution iterative process of GWO is nonlinear and highly complicated, the linearly decrease transition rule of a cannot truly reflect the actual optimization process. Additionally, the aim of this study is to utilize longer time for local search process as compared to the global search process. Therefore, this paper presents a novel nonlinear decrease transition rule of a and is given by:

$$a(t) = a_{\text{start}} - (a_{\text{start}} - a_{\text{end}}) \cdot \cos\left(\left(1 - \frac{t}{T}\right) \times \frac{\pi}{2}\right) \quad (1)$$

where a_{start} and a_{end} are the initial and end values of the transition parameter a, t indicates the iterative number, and T denotes the maximum iterative number.

Compared with the original linear strategy described by Eq. (4) in the basic GWO, the nonlinear decrease transition rule of a described by Eq. (9) is more concentrated on the local search. Figure 2 shows the comparison graph of the proposed nonlinear transition rule and the original linear strategy.

From Figure 2 and Eq. (4), in the basic GWO, half of the iterations numbers are used to global search $(a > 1)$, and the other half of the iterations numbers are focused on local search $(a < 1)$ [22]. As seen from Figure 2 and Eq. (9), the values of the presented nonlinear

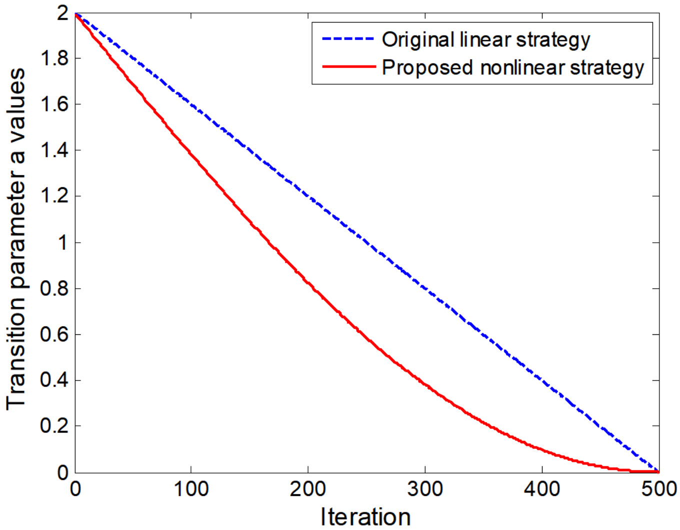

Figure 2. Comparison of two transition parameters.

transition parameter a are small $(a < 1)$ in the middle and late stages of search, which indicates that it is focused on the local search for a long time (about 66% of the total iteration numbers) as compared to the global search process. Figure 2 also demonstrates that the presented nonlinear transition parameter a is concentrated on the global search only for about 34% iterations.

3.2 Cuckoo search algorithm

The cuckoo search (CS) algorithm is an effective nature-inspired meta-heuristic optimization technique which simulates the obligate brood parasitism characteristics of cuckoos [45]. In the CS algorithm, a cuckoo indicates a individual of the search space. In the search process of CS, a new individual is produced by using Lévy flight (LF) operator [46]:

$$X_i(t+1) = X_i(t) + \gamma \oplus Levy(\lambda) \quad (10)$$

$$\gamma = \gamma_0 \cdot (X_i(t) - X_i(t)) \quad (11)$$

where $X(t)$ represents the current individual at t generation, γ_0 denotes a constant number in [0.001 0.01], $\dot{X}_i(\#)$ indicates the randomly selected individual from the population, $\oplus$ represents the entry wise multiplication, the Lévy(λ) is a random walk, in which the random step size S is derived from a Lévy distribution, and the λ indicates the Lévy flight exponent:

$$Levy(\lambda) = \frac{\lambda \cdot \Gamma(\lambda)\sin(\pi\lambda/2)}{\pi} \cdot \frac{1}{S^{1+\lambda}} \quad (12)$$

$$S = \frac{u}{|v|^{1/\lambda}} \quad (13)$$

where u and v are the constant numbers and obey with Gaussian normal distribution.

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \tag{14}$$

$$\sigma_u = \left[\frac{\Gamma(1+\lambda)\cdot\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}\cdot\Gamma(1+\lambda)/2!\cdot\lambda}\right]^{1/\lambda}, \quad \sigma_v = 1 \tag{15}$$

where $\Gamma(\cdot)$ represents the standard Gamma function.

A fraction P_a (abandon probability) of the worst individuals is abandoned and new individuals are produced.

$$X_i(t+1) = X_i(t) + \gamma_0 \oplus H(P_a - r) \oplus (X_i(t) - X_k(t)) \tag{16}$$

where r is a random number, H is a Heaviside function, $\dot{X}_k$ is randomly selected individual from population.

Algorithm 2 outlines the pseudo code of the original CS algorithm.

Algorithm 2: Cuckoo search (CS) algorithm

```
Initialize the population of nests
Initialize the parameters
Calculate the fitness of each individual
Determine the best nest (X_max) and the best
  fitness (F_max)
t ← 0
while t < T do
    Generate a new solution by using Eq. (10)
    Evaluate the fitness of each individual
    Select randomly two individuals (say j and k)
    A fraction P_a of worse nests are abandoned
      and new nests are generated by using Eq. (16)
    Update the best nest and the best fitness
    t ← t + 1
end
Return X_max and F_max
```

3.3 Adaptive position search equation

The meta-heuristic method is designed to achieve a tradeoff between convergence and diversity in the process of global optimization. This tradeoff is very crucial to the successful execution of optimization technique. In the basic GWO, the other wolves in the population are attracted by α, β, and δ wolves, and they may easily fall into the local optima due to the lack of diversity among individuals and the over-learning from the global optimal individual found to date. Based on the position search equation of GWO shown in Eq. (8), a new candidate search agent is produced by moving the position of the current wolf toward the α, β, and δ wolves. Thus, the position search equation described by Eq. (8) is beneficial to convergence but the diversity is poor [23, 24]. Many attempts have also been suggested that the modified position search equation is one of the effective techniques for improving the search performance of GWO [19]. Different from the previous works, this paper designed a novel adaptive position search equation and the detailed explanations are as follows.

As seen from Eq. (8), the new candidate search agent is produced by between the global optimal solution (α wolf) and the old one. This position search equation may be efficient in earlier iterations. The reasons are as follows. In general, the population has a good diversity in the early stage of evolution search. The good diversity means that the population has strong capability to explore the entire search space [19]. Accelerate convergence is one of the main objective of this stage. The Eq. (8) can achieve this goal. Moreover, in the original GWO, the personal historical best information ($pbest$) has not been fully utilized [23]. Consequently, in this paper, the new candidate individual generated by using the global best individual (X) and the personal historical best ($pbest$), and design a novel self-adaptive position search equation, i.e.,

$$X(t+1) = b\times(Y_1+Y_2+Y_3)/3+(1-b)\times(X_{pbest}(t)-X(t)) \tag{17}$$

where X_{pbest} represents the personal historical best position of X, and $b \in [0, 1]$ is a control parameter.

To convert dynamically the search ability of Eq. (17), the control parameter b is calculated by:

$$b(t) = 1 - \frac{t}{T} \tag{18}$$

where T represents the total iteration numbers.

Compared with the position search Eq. (8) in the original GWO algorithm, the proposed position search Eq. (17) has two different features: 1) The individual learns not only from its own information but also from the information of the other individual in the population; and 2) The control parameter b is introduced to adjust adaptive the search performance of the Eq. (17).

In summary, the flow chart of EGWO algorithm is provided in Figure 3.

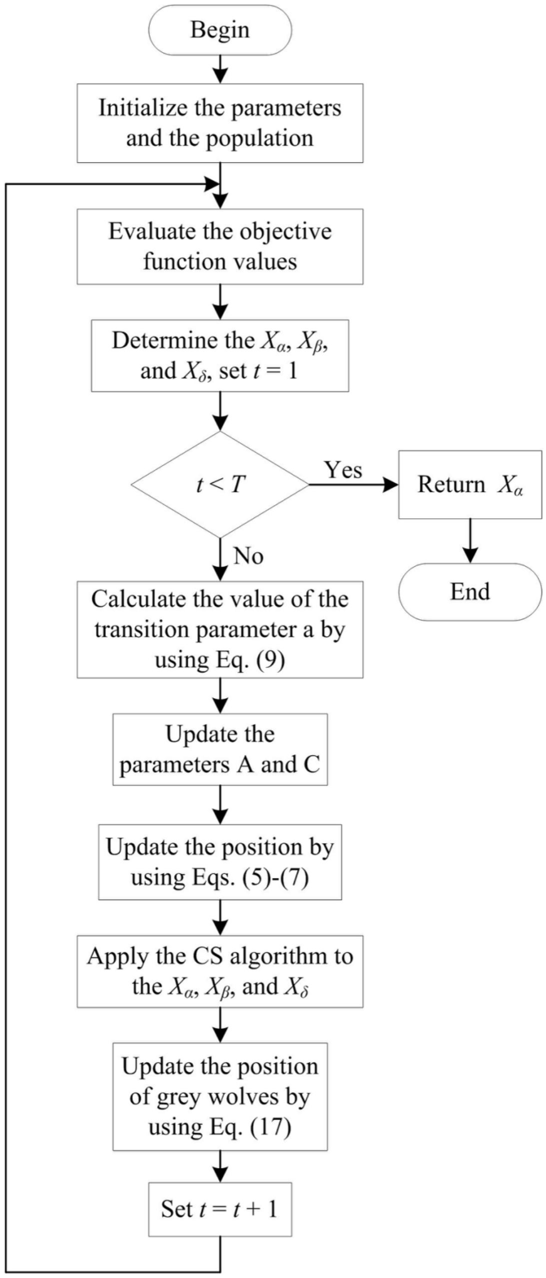

Figure 3. The flow chart of the proposed EGWO algorithm.

4 Experimental Results

To investigate the effectiveness and feasibility of EGWO, a set of experiments are conducted. Firstly, the EGWO is applied to solve the classical benchmark test functions. Then, engineering design problems are applied to further verify the overall performance. Finally, EGWO is utilized for solving the feature selection problems.

4.1 EGWO for classical benchmark test functions

In this subsection, 25 classical benchmark test functions are used for experiments. The detailed characteristics of 25 functions are given in Table 1. These selected problems can be divided into two categories: unimodal and multimodal functions. The unimodal functions (f_1-f_{12}) have only one global optimal solution and no local optimal solution. They are suitable to test the local exploitation capability of algorithm. Conversely, multimodal functions (f_{13}-f_{25}) are usually utilized to investigate the global search ability of algorithm since they have many local optimal solutions [19].

4.1.1 Compared with the basic GWO algorithm

The results of EGWO are compared with the basic GWO algorithm on low dimensional classical benchmark functions. For two algorithms, the population number is $N = 30$ and the total iteration number is 500. The other parameters of two algorithms are provided: in GWO, a is linearly decreased from 2 to 0; in EGWO, $a_{\text{start}} = 2$, $a_{\text{end}} = 0$, b is linearly decreased from 1 to 0.

Table 2 shows the best, mean, worst, and standard deviation (St.dev) values obtained by EGWO and GWO on 25 classical benchmark problems in Table 1. The dimension of each function is set to 30. Two algorithms independently run 30 times to reduce the error.

To ensure experimental reproducibility, the main parameter settings of all compared algorithms are summarized as follows:

For WPSO, the population size $N = 30$, inertia weight ω decreases linearly from 0.9 to 0.4, and acceleration coefficients $c_1 = c_2 = 2.0$. For CMAES, the population size is set to $4 + 3\log(D)$ and the learning rate parameters follow its standard implementation. For ODE, the scaling factor $F = 0.5$ and crossover rate $CR = 0.9$. For WOA and HHO, the control coefficients are set according to their original papers. For mGWO and MGWO, the nonlinear coefficient and memory update rule are used as defined in their respective publications. All algorithms are executed with the same population size and iteration number to ensure fairness of comparison.

As seen from Table 2, EGWO obtains the theoretical optima (0) for eleven test problems (i.e., f_5, f_6, f_9, f_{12}, f_{13}, f_{15}, f_{18}, f_{19}, f_{22}, f_{23}, and f_{25}). Compared with the original GWO algorithm, EGWO finds the better results on 20 benchmark test functions. For f_5, f_{15}, f_{18}, f_{22}, and f_{23}, the similar results are obtained by EGWO and GWO. In addition, Figure 4 plots the convergence graphs of EGWO and GWO on twelve representative problems with 30 dimensions. From Figure 4, compared with the basic GWO algorithm, EGWO gets higher solution accuracy and faster

Table 1. The 25 classical benchmark test functions.

Function equation	Domain	f_{min}
$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0
$f_2(x) = \sum_{i=1}^{D} \|x_i\| + \prod_{i=1}^{D} \|x_i\|$	$[-10, 10]^D$	0
$f_3(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$	0
$f_4(x) = \max\{\|x_i\|, 1 \leq i \leq D\}$	$[-100, 100]^D$	0
$f_5(x) = \sum_{i=1}^{D}(x_i + 0.5)^2$	$[-100, 100]^D$	0
$f_6(x) = \sum_{i=1}^{D} ix_i^4$	$[-1.28, 1.28]^D$	0
$f_7(x) = \sum_{i=1}^{D} ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]^D$	0
$f_8(x) = \sum_{i=1}^{D} -x_i^2$	$[-10, 10]^D$	0
$f_9(x) = \sum_{i=1}^{D} \|x_i\|^{i+1}$	$[-1, 1]^D$	0
$f_{10}(x) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} x_i^2$	$[-100, 100]^D$	0
$f_{11}(x) = x_1^2 + 10^6 \cdot \sum_{i=2}^{D} x_i^6$	$[-100, 100]^D$	0
$f_{12}(x) = 10^6 \cdot x_1^2 + \sum_{i=2}^{D} x_i^6$	$[-1, 1]^D$	0
$f_{13}(x) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	0
$f_{14}(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	$[-32, 32]^D$	0
$f_{15}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$	0
$f_{16}(x) = \sum_{i=1}^{D}[x_i \cdot \sin(x_i) + 0.1 \cdot x_i]$	$[-10, 10]^D$	0
$f_{17}(x) = \sin^2(\pi x_1) + \sum_{i=1}^{D-1}[x_i^2 \cdot (1 + 10\sin^2(\pi x_1)) + (x_i - 1)^2 \cdot \sin^2(2\pi x_i)]$	$[-10, 10]^D$	0
$f_{18}(x) = 0.1D - (0.1\sum_{i=1}^{D}\cos(5\pi x_i) - \sum_{i=1}^{D} x_i^2)$	$[-1, 1]^D$	0
$f_{19}(x) = \sum_{i=2}^{D}\left(0.5 + \frac{\sin^2(\sqrt{100x_{i-1}^2 + x_i^2}) - 0.5}{1 + 0.001(x_{i-1}^2 - 2x_{i-1}x_i + x_i^2)^2}\right)$	$[-100, 100]^D$	0
$f_{20}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2(1 + \sin^2(2\pi x_D)))$	$[-5, 5]^D$	0
$f_{21}(x) = \sum_{i=1}^{D}(0.2x_i^2 + 0.1x_i^2 \cdot \sin(2x_i))$	$[-10, 10]^D$	0
$f_{22}(x) = (-1)^{D+1}\prod_{i=1}^{D}\cos(x_i) \cdot \exp[-\sum_{i=1}^{D}(x_i - \pi)^2]$	$[-100, 100]^D$	0
$f_{23}(x) = \sum_{i=1}^{D}[x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7]$	$[-15, 15]^D$	0
$f_{24}(x) = \sum_{i=1}^{D-1}(x_i^2 + 2x_{i+1}^2)^{0.25} \cdot ((\sin 50(x_i^2 + x_{i+1}^2)^{0.1})^2 + 1)$	$[-10, 10]^D$	0
$f_{25}(x) = \sum_{i=1}^{D} x_i^6 \cdot (2 + \sin\frac{1}{x_i})$	$[-1, 1]^D$	0

convergence speed.

4.1.2 Scalability test

Moreover, the EGWO is applied for solving the 25 classical benchmark test functions with higher dimensions (i.e., 100 and 1000 dimensions) to further investigate its scalability. In this experiment, the same parameter settings as before are used. The best, mean, worst, st.dev results of EGWO and GWO on 25 problems with high dimensionality are outlined in Tables 3 and 4.

From Tables 3 and 4, EGWO shows quite excellent scalability for the search dimensions on all the functions in Table 1. That is to say, the overall performance of the proposed EGWO does not seriously deteriorate. It must be emphasized that a function with 1000 dimensions is very challenging for GWO.

Table 2. Comparisons of EGWO and GWO on 25 classical problems with 30 dimensions in Table 1.

Function	GWO				EGWO			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
f_1	4.18E-28	1.07E-27	2.30E-26	1.00E-26	1.42E-202	1.37E-201	3.90E-201	0
f_2	3.37E-17	6.23E-17	9.49E-17	2.70E-17	5.19E-103	2.08E-102	5.75E-102	2.20E-102
f_3	1.04E-06	2.98E-05	1.02E-04	4.21E-05	7.30E-185	1.46E-183	6.39E-183	0
f_4	1.81E-07	6.61E-07	2.05E-06	7.89E-07	1.31E-97	2.42E-96	5.82E-96	1.29E-96
f_5	0	0	0	0	0	0	0	0
f_6	1.65E-53	3.02E-52	5.90E-52	2.41E-52	0	0	0	0
f_7	1.22E-03	2.13E-03	2.83E-03	6.08E-04	1.02E-05	6.48E-05	1.02E-04	3.80E-05
f_8	8.83E-30	9.37E-29	5.59E-28	2.29E-28	1.18E-203	9.67E-202	4.26E-201	0
f_9	1.45E-106	2.08E-98	1.04E-97	4.65E-98	0	0	0	0
f_{10}	4.03E-25	1.09E-24	1.70E-24	4.91E-25	7.08E-200	4.38E-198	1.69E-197	0
f_{11}	1.15E-53	2.76E-51	5.70E-51	2.62E-51	9.87E-294	1.05E-287	5.21E-287	0
f_{12}	1.13E-69	3.96E-65	2.91E-64	1.27E-64	0	9.24E-299	1.35E-298	0
f_{13}	5.68E-14	1.91E+00	5.95E+00	2.75E+00	0	0	0	0
f_{14}	7.90E-14	9.97E-14	1.15E-13	1.73E-14	8.88E-16	8.88E-16	8.88E-16	0
f_{15}	0	0	0	0	0	0	0	0
f_{16}	2.21E-16	2.07E-04	9.04E-04	3.94E-04	6.40E-104	1.39E-95	6.96E-95	3.11E-95
f_{17}	5.19E-30	5.26E-29	2.15E-28	8.95E-29	9.74E-204	2.09E-203	6.52E-203	0
f_{18}	0	0	0	0	0	0	0	0
f_{19}	1.05E+01	1.12E+01	1.16E+01	4.57E-01	0	0	0	0
f_{20}	1.51E-32	2.32E-31	8.07E-31	3.24E-31	2.40E-206	4.78E-205	1.40E-204	0
f_{21}	2.37E-31	1.34E-30	2.55E-30	9.07E-31	2.94E-204	2.53E-203	8.00E-203	0
f_{22}	0	0	0	0	0	0	0	0
f_{23}	0	0	0	0	0	0	0	0
f_{24}	1.59E-07	5.24E-07	8.23E-07	2.45E-07	7.96E-51	1.35E-50	2.10E-50	5.17E-51
f_{25}	2.13E-69	2.36E-64	5.38E-64	2.37E-64	0	0	0	0

The reason is that it does not use the specific search strategies customized to deal with high-dimensional optimization problems. Compared with the low dimensions functions (i.e., 30 dimensions), the overall performance of the original GWO has been deteriorated on most problems with high dimensionality (100 and 1000 dimensions). In other words, the original GWO algorithm cannot effectively and efficiently deal with the high dimensional optimization problems. Meanwhile, the search performance of the proposed EGWO is significantly better than the original GWO on all the functions with 100 and 1000 dimensions except for f_5 and f_{22}. The similar results are obtained by EGWO and GWO on functions f_5 and f_{22}.

Moreover, Figures 5 and 6 plot the convergence graphs of EGWO and GWO on 12 representative benchmark problems with 100 and 1000 dimensions, respectively. As can be seen from Figures 5 and 6, EGWO shows very better performance than GWO on all the 12 representative functions for solution accuracy and convergence speed.

4.1.3 *Compared with the other algorithms*

To further investigate its effectiveness, the proposed EGWO is applied to compare with the other algorithms such as Weighted PSO (WPSO) [47], Covariance Matrix Adaptation Evolution Strategy (CMAES) [48],

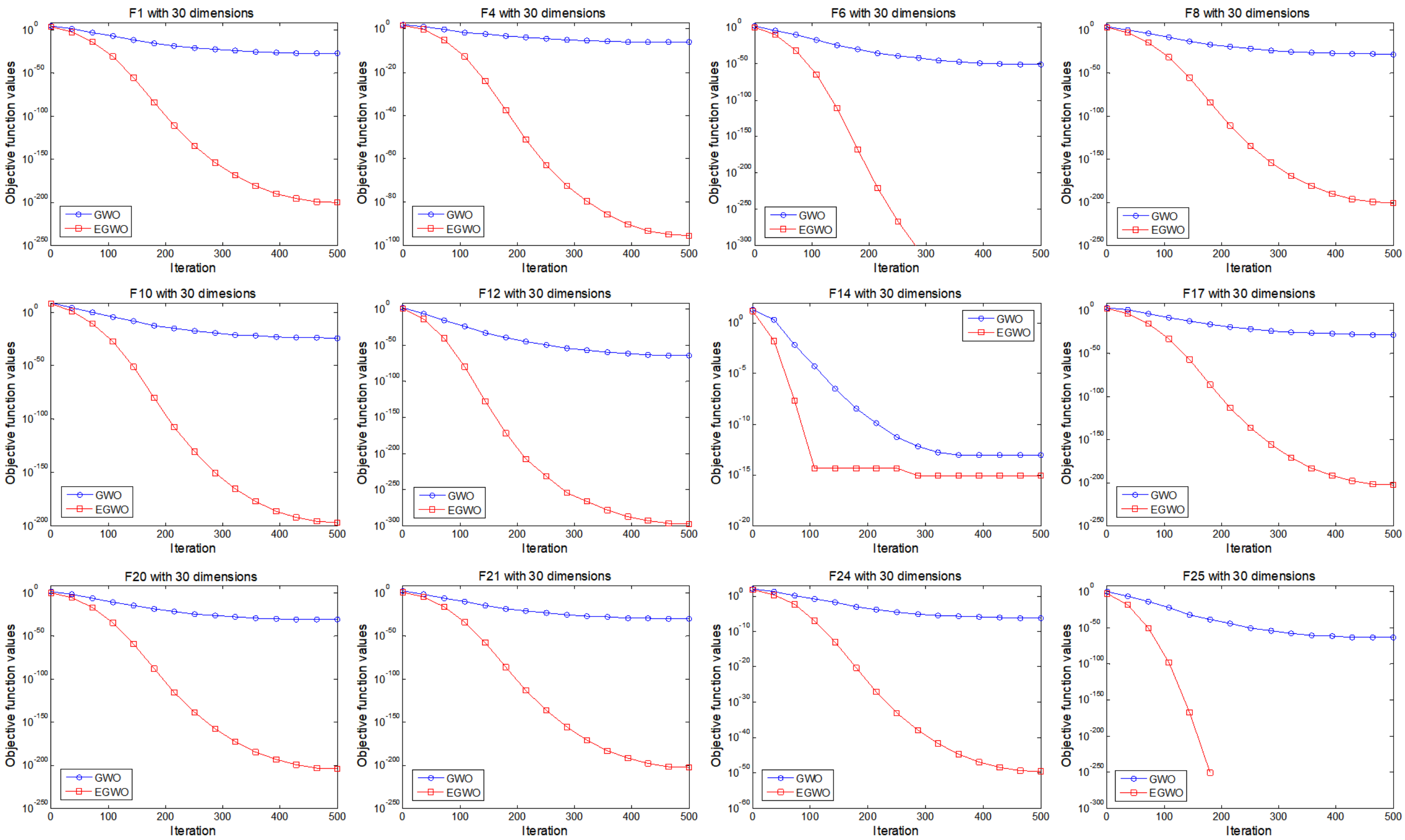

Figure 4. Convergence graphs of EGWO and GWO on 12 representative functions with 30 dimensions.

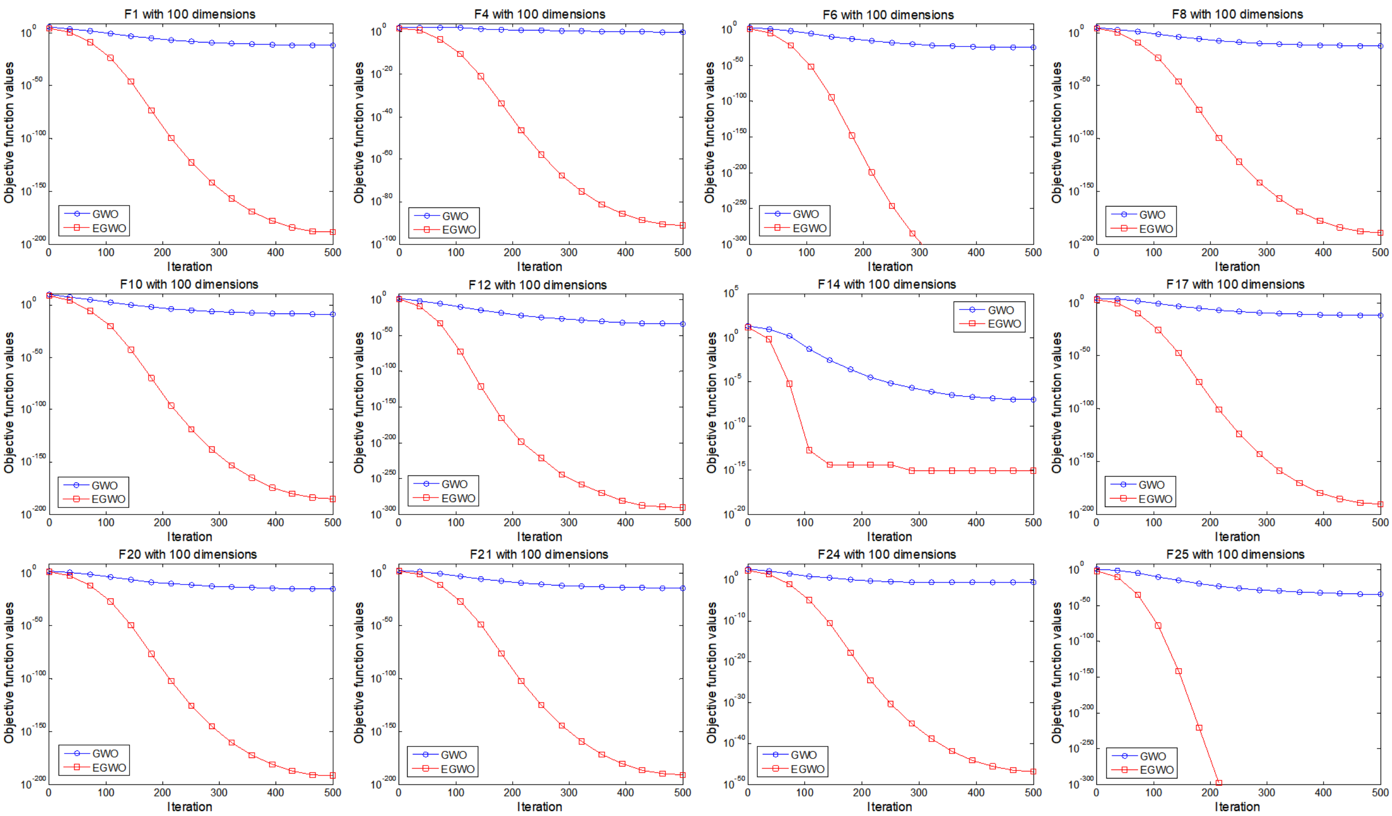

Figure 5. Convergence graphs of EGWO and GWO on 12 representative functions with 100 dimensions.

Opposition-based DE (ODE) [49], whale optimization algorithm (WOA) [50], Harris hawks optimizer

Table 3. Comparisons of EGWO and GWO on 25 classical benchmark functions with 100 dimensions in Table 1.

Function	GWO				EGWO			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
f_1	2.77E-13	1.43E-12	3.06E-12	1.25E-12	1.93E-190	1.88E-189	3.28E-189	0
f_2	2.27E-08	4.02E-08	7.20E-08	1.79E-08	6.39E-97	9.07E-97	1.43E-96	3.45E-97
f_3	7.18E+01	3.87E+02	1.13E+03	4.32E+02	7.26E-177	1.49E-175	4.48E-175	0
f_4	7.85E-02	8.35E-01	1.77E+00	6.70E-01	2.97E-92	8.89E-92	1.80E-91	5.67E-92
f_5	0	0	0	0	0	0	0	0
f_6	1.03E-26	1.58E-25	4.21E-25	1.79E-25	0	0	0	0
f_7	2.40E-03	5.99E-03	7.93E-03	2.14E-03	2.29E-05	3.36E-05	4.90E-05	9.86E-06
f_8	5.41E-14	4.60E-13	1.22E-12	4.94E-13	4.61E-191	1.54E-189	4.17E-189	0
f_9	2.60E-79	4.32E-62	2.16E-61	9.66E-62	0	0	0	0
f_{10}	1.09E-09	2.46E-09	3.86E-09	7.44E-10	4.83E-187	8.78E-186	1.28E-185	0
f_{11}	3.02E-19	1.33E-17	4.04E-17	1.83E-17	1.46E-279	3.03E-273	1.43E-272	0
f_{12}	4.35E-37	4.13E-34	1.75E-33	7.81E-34	0	2.94E-291	1.88E-290	0
f_{13}	6.74E+00	1.09E+01	1.57E+01	3.73E+00	0	0	0	0
f_{14}	6.86E-08	9.21E-08	1.22E-07	2.46E-08	8.88E-16	8.88E-16	8.88E-16	0
f_{15}	4.23E-13	8.80E-03	2.73E-02	1.26E-02	0	0	0	0
f_{16}	6.45E-04	3.40E-03	8.90E-03	3.18E-03	1.25E-97	2.84E-93	1.42E-92	6.35E-92
f_{17}	1.60E-13	1.67E-12	5.45E-12	2.30E-12	3.01E-192	5.80E-191	1.10E-190	0
f_{18}	2.13E-14	3.09E-14	3.91E-14	7.71E-15	0	0	0	0
f_{19}	4.50E+01	4.52E+01	4.54E+01	1.56E-01	0	0	0	0
f_{20}	2.39E-16	7.03E-16	2.00E-15	7.08E-16	3.42E-193	1.68E-192	3.79E-192	0
f_{21}	1.11E-15	6.88E-15	1.34E-14	5.07E-15	1.94E-192	8.63E-192	1.34E-191	0
f_{22}	0	0	0	0	0	0	0	0
f_{23}	2.95E-13	1.29E-12	2.67E-12	1.03E-12	0	0	0	0
f_{24}	1.73E-01	2.85E-01	3.99E-01	8.20E-02	9.38E-48	1.62E-47	2.02E-47	4.54E-48
f_{25}	1.29E-36	3.04E-34	8.76E-34	3.60E-34	0	0	0	0

(HHO) [51], modified GWO (mGWO) [22], and Memory GWO (MGWO) [52]. The parameter settings of these methods are the same as their corresponding literatures. In these algorithms, the WPSO, CMAES, and ODE are the well-known meta-heuristic optimization techniques, while the WOA, HHO, mGWO, and MGWO are the relatively new meta-heuristic methods. In this experiment, 25 classical problems are chosen from Table 1 to verify the performance of the compared algorithms. The dimensions of the selected problems are 30. To a fair comparison, the population number is 30, and the total iterative number is 500. The statistical results (mean and st.dev) of eight algorithms on 25 classical benchmark functions are given in Table 5. Moreover, according to the average results of each problem, the Wilcoxon's rank sum test results of EGWO and the selected methods are also outlined in Table 5. The symbol of "+", "≈", and "-" indicate that the overall performance of EGWO is superior to, similar to, and inferior to the corresponding algorithms in Table 5. The best result of each function is highlighted in bold.

As seen in Table 5, compared with WPSO and CMAES, EGWO obtains the better mean and st.dev values on all the functions except for f_5 and f_{22}. The similar results are obtained by EGWO, WPSO, and CMAES algorithms on two functions (i.e., f_5 and f_{22}). Compared to ODE, EGWO gets the better

Table 4. Comparisons of EGWO and GWO on 25 classical benchmark functions with 1000 dimensions in Table 1.

Function	GWO				EGWO			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
f_1	1.85E-01	2.24E-01	2.47E-01	2.43E-02	1.23E-180	8.07E-180	2.23E-179	0
f_2	3.62E-01	5.20E-01	7.79E-01	1.66E-01	1.03E-92	4.41E-92	3.54E-92	1.45E-92
f_3	6.35E+05	1.35E+06	2.05E+06	5.32E+05	3.42E-168	1.78E-167	3.23E-167	0
f_4	7.41E+01	7.66E+01	7.87E+01	1.91E+00	5.13E-88	2.67E-87	8.35E-87	3.14E-87
f_5	0	0	0	0	0	0	0	0
f_6	3.11E-06	6.02E-06	9.58E-06	2.62E-06	0	0	0	0
f_7	1.20E-01	1.65E-01	2.48E-01	5.27E-02	2.20E-06	5.67E-05	1.49E-04	5.82E-05
f_8	7.20E-01	9.57E-01	1.49E+00	3.04E-01	8.72E-181	8.04E-180	2.35E-179	0
f_9	3.44E-06	6.09E-03	2.42E-02	1.03E-02	0	0	0	0
f_{10}	5.34E+02	7.17E+02	9.10E+02	1.34E+02	1.75E-177	6.69E-177	2.04E-176	0
f_{11}	1.69E+07	2.38E+08	9.58E+08	4.08E+08	6.56E-266	1.12E-261	5.19E-261	0
f_{12}	2.60E-11	1.59E-10	6.25E-10	2.49E-10	1.22E-284	3.12E-279	8.98E-279	0
f_{13}	1.73E+02	1.90E+02	2.22E+02	2.07E+02	0	0	0	0
f_{14}	1.64E-02	1.73E-02	2.34E-02	1.90E-03	8.88E-16	8.88E-16	8.88E-16	0
f_{15}	1.39E-12	1.69E-02	1.85E-02	1.80E-03	0	0	0	0
f_{16}	2.48E-01	4.58E-01	1.25E+00	4.43E-01	3.70E-93	7.91E-93	1.23E-92	3.34E-93
f_{17}	1.26E+01	3.43E+02	1.78E+03	6.87E+02	1.81E-182	7.45E-182	1.91E-181	0
f_{18}	9.81E-05	1.83E-04	2.87E-04	7.43E-05	0	0	0	0
f_{19}	4.85E+02	4.87E+02	4.89E+02	1.62E+00	0	0	0	0
f_{20}	1.65E-04	7.43E-04	2.40E-03	9.60E-04	1.73E-184	6.76E-184	1.22E-183	0
f_{21}	6.03E-04	1.13E-03	1.77E-03	4.73E-04	1.42E-183	1.04E-182	1.46E-182	0
f_{22}	0	0	0	0	0	0	0	0
f_{23}	2.67E-01	4.79E+00	1.28E+01	4.78E+00	0	0	0	0
f_{24}	7.27E+01	8.24E+01	9.25E+01	5.90E+00	2.30E-45	3.49E-45	4.94E-45	1.07E-45
f_{25}	6.00E-10	5.81E-09	1.08E-08	4.32E-09	0	0	0	0

and similar values on 21 and four problems (i.e., f_5, f_{15}, f_{22}, and f_{23}), respectively. The performance of EGWO is much better than WOA and MGWO for 19 problems. In addition, the similar results are provided by EGWO, WOA, and MGWO on functions f_5, f_{13}, f_{15}, f_{18}, f_{22}, and f_{23}. The overall performance of EGWO is superior to the HHO algorithm on seventeen benchmark functions. For f_5, f_{13}-f_{15}, f_{18}, f_{22}, f_{23}, and f_{25} functions, EGWO and HHO find the similar results. The search performance of EGWO is much better than mGWO on 20 benchmark functions. However, the similar results are provided on five test functions (i.e., f_5, f_{15}, f_{18}, f_{22}, and f_{23}).

Moreover, the total Friedman's rank results of each algorithm for each function are shown in Figure 7 according to the average errors of the fitness function values. From Figure 7, the EGWO obtains the first rank, followed by the HHO, MGWO, WOA, mGWO, ODE, CMAES, and WPSO.

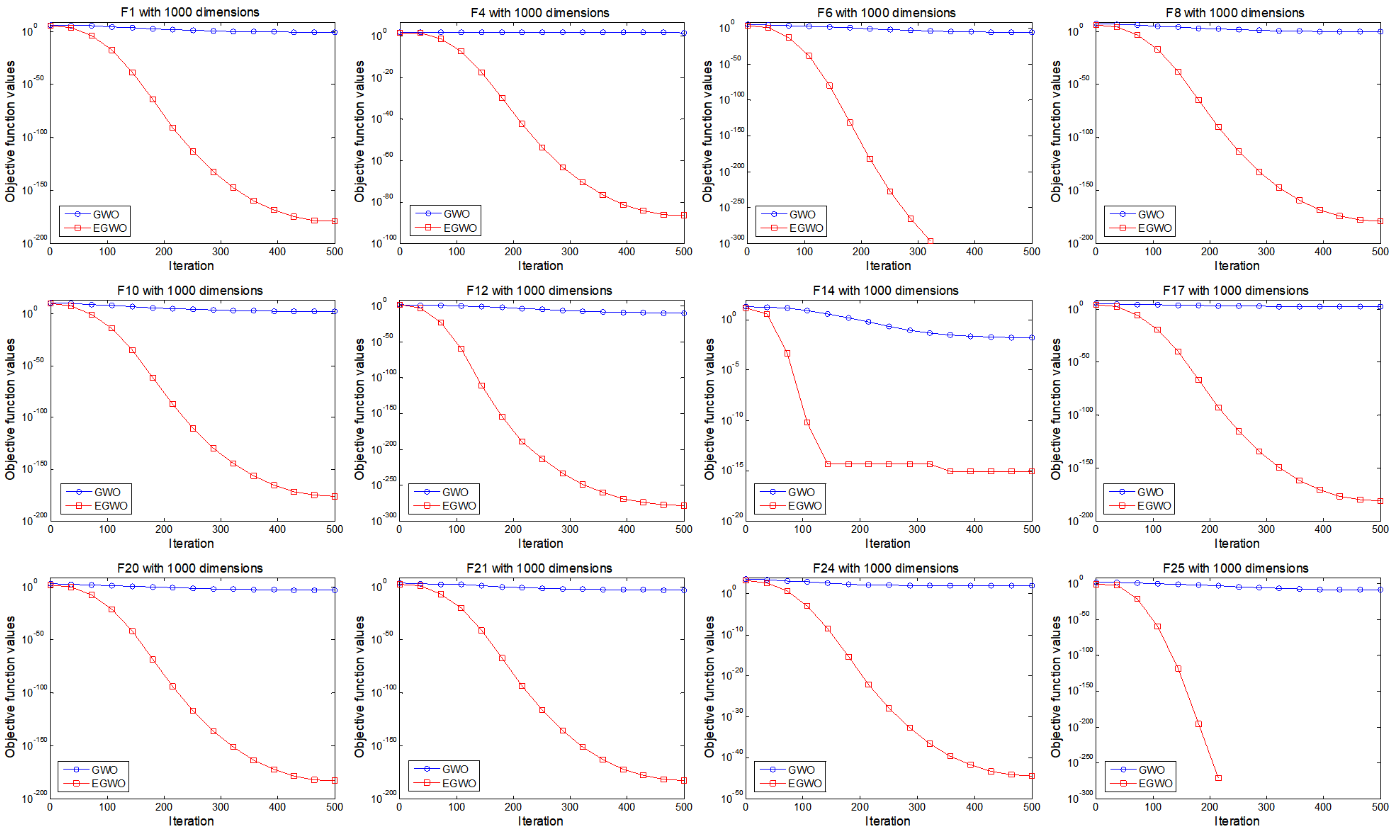

Figure 6. Convergence graphs of EGWO and GWO on 12 representative functions with 1000 dimensions.

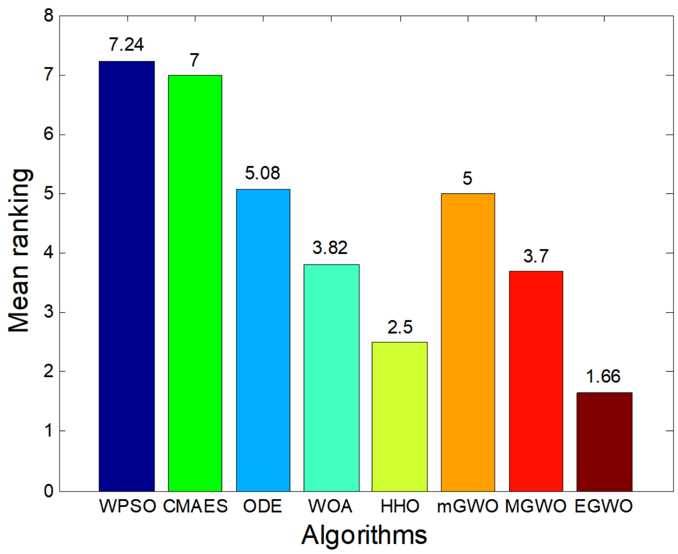

Figure 7. Mean Friedman test ranks of eight algorithms on 25 benchmark functions with 30 dimensions.

4.2 EGWO for engineering optimization problems

Although EGWO has shown excellent performance on benchmark functions, it is necessary to investigate its effectiveness on engineering problems. Thus, three well-known engineering design problems are used to verify the effectiveness of EGWO. These problems are the constrained optimization problems. How to handle constraints is an important issue when applying EGWO to solve these engineering optimization problems. This paper introduces the Deb's feasibility-rule [53] to deal with constraints.

4.2.1 Compression/tension spring design problem

This problem considers a compression/tension spring structure which is shown in Figure 8. The goal of this problem is to obtain the minimum weight of a spring subject to three nonlinear constraints. This problem has three decision variables namely wire diameter $d(x_1)$, coil diameter $D(x_2)$, and the number of active coils $P(x_3)$.

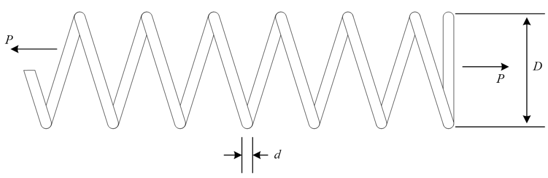

Figure 8. The structure of compression/tension spring design problem.

The formulation of this problem is defined as

minimize $f(X) = (x_3 + 2)x_2 x_1^2, X = [x_1, x_2, x_3]$

Table 5. Comparisons of EGWO and other seven algorithms on 25 classical functions in Table 1.

Function	Index	WPSO	CMAES	ODE	WOA	HHO	mGWO	MGWO	EGWO
f_1	Mean	3.19E-04	1.73E-05	7.10E-36	1.59E-73	1.14E-99	2.30E-36	6.02E-54	1.37E-201
	Std	2.22E-04	5.49E-06	8.58E-36	2.70E-73	2.50E-99	3.57E-36	5.75E-54	0
f_2	Mean	3.96E-03	6.03E-03	3.88E-20	1.46E-50	5.72E-52	7.32E-22	9.46E-32	2.08E-102
	Std	2.33E-03	2.93E-03	4.74E-20	3.26E-50	1.17E-51	3.37E-22	6.90E-32	2.20E-102
f_3	Mean	5.05E-04	3.10E-05	4.85E-36	5.19E-04	5.13E-82	5.29E-06	2.00E-16	1.46E-183
	Std	4.92E-04	1.96E-05	8.63E-36	2.02E-04	1.13E-81	1.18E-05	4.37E-16	0
f_4	Mean	3.45E+00	3.31E-02	5.85E-13	4.36E-01	4.77E-51	1.38E-09	1.85E-17	2.42E-96
	Std	5.88E-01	7.06E-02	3.55E-13	2.84E-01	1.01E-50	3.53E-10	6.95E-18	1.29E-96
f_5	Mean	0	0	0	0	0	0	0	0
	Std	0	0	0	0	0	0	0	0
f_6	Mean	5.90E-11	8.32E-16	1.07E-67	2.64E-12	1.23E-19	1.82E-63	1.43E-91	0
	Std	8.15E-11	1.06E-15	2.38E-67	5.74E-12	0	3.11E-63	3.19E-91	0
f_7	Mean	2.13E-02	1.25E-01	3.88E-03	2.31E-03	6.87E-05	1.74E-03	6.02E-04	6.48E-05
	Std	9.29E-03	1.36E-01	1.35E-03	2.71E-03	7.19E-05	8.44E-04	2.13E-04	3.80E-05
f_8	Mean	4.91E-05	1.43E-05	1.21E-37	7.60E-77	2.52E-98	2.22E-37	2.01E-53	9.67E-202
	Std	2.02E-05	1.09E-05	1.76E-37	1.69E-76	5.63E-98	1.53E-37	4.01E-53	0
f_9	Mean	1.68E-23	3.87E-08	3.00E-11	1.16E-10	1.75E-12	7.70E-12	3.32E-18	0
	Std	2.21E-23	2.53E-08	6.71E-11	2.19E-10	2.70E-12	1.70E-12	0	0
f_{10}	Mean	1.24E+00	1.20E-06	5.55E-33	2.42E-72	6.51E-98	5.63E-33	9.18E-51	4.38E-198
	Std	9.96E-01	4.01E-05	7.72E-33	5.41E-72	1.30E-97	7.30E-33	8.64E-51	0
f_{11}	Mean	2.22E-03	2.88E-01	2.17E-76	2.38E-104	2.62E-133	1.61E-64	1.76E-109	1.05E-287
	Std	1.17E-03	4.49E-01	2.03E-76	5.32E-104	5.86E-133	3.58E-64	3.80E-109	0
f_{12}	Mean	3.19E-13	1.54E+00	8.02E-90	2.72E-109	2.88E-138	6.65E-82	2.23E-122	9.24E-299
	Std	5.64E-13	3.52E-01	1.79E-89	6.08E-109	6.44E-138	1.40E-81	4.50E-122	0
f_{13}	Mean	3.98E+01	9.17E-01	3.55E-01	0	0	3.42E-14	0	0
	Std	7.25E+00	8.68E-01	1.15E-01	0	0	5.10E-14	0	0
f_{14}	Mean	4.88E-03	1.32E-03	2.22E-15	3.73E-15	8.88E-16	2.43E-14	7.99E-15	8.88E-16
	Std	2.51E-03	3.60E-04	0	2.97E-15	0	4.78E-15	0	0
f_{15}	Mean	6.01E-04	9.79E-06	0	0	0	0	0	0
	Std	6.46E-04	3.72E-06	0	0	0	0	0	0
f_{16}	Mean	1.61E-03	1.35E-03	1.37E-03	8.51E-51	5.12E-39	2.95E-05	2.66E-27	1.39E-95
	Std	1.55E-03	4.68E-04	3.07E-03	1.88E-50	2.15E-38	4.06E-05	5.95E-27	3.11E-95
f_{17}	Mean	7.92E-01	5.94E-01	5.09E-37	5.32E-76	2.13E-98	1.97E-38	1.04E-54	2.09E-203
	Std	8.28E-01	8.85E-01	7.67E-37	1.18E-75	3.08E-98	3.37E-38	1.21E-54	0
f_{18}	Mean	2.96E-01	6.53E-08	1.35E-40	0	0	0	0	0
	Std	2.76E-01	2.49E-08	1.77E-40	0	0	0	0	0
f_{19}	Mean	1.03E-01	1.30E-01	4.10E+00	2.06E-01	1.77E-05	1.10E-01	8.88E+00	0
	Std	2.75E-01	1.73E-01	3.18E-01	4.44E-01	2.71E-05	4.29E-01	1.25E+00	0
f_{20}	Mean	7.89E-07	8.32E-07	2.78E-39	1.46E-78	7.51E-11	1.26E-40	1.26E-58	4.78E-205
	Std	1.50E-06	4.24E-07	3.96E-39	3.17E-78	1.12E-10	1.80E-40	2.80E-58	0

s.t.

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \leq 0$$

$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0.$$

where $0.25 \leq x_1 \leq 1.30, 0.05 \leq x_2 \leq 2, 2.00 \leq x_3 \leq 15$.

The parameter settings of EGWO are given as follows. The population size is 50 and the maximum number of iteration is 5000. The results obtained by EGWO and other optimization techniques such as CMAES [48], JADE [54], CPSO [55], GWO [3], mGWO [22], EEGWO [19], IGWO [23], MGWO [52], and ISCA [56] are outlined in Table 6. From Table 6, compared with the other selected algorithms, EGWO and ISCA obtain minimum weight for compression/tension spring design problem.

4.2.2 Three-bar truss design problem

As given in Figure 9, this problem considers a three-bar planar truss structure. The purpose of this case is to obtain the minimum volume of a three-bar truss with three inequality constraints. This design problem has two decision variables (x_1 and x_2).

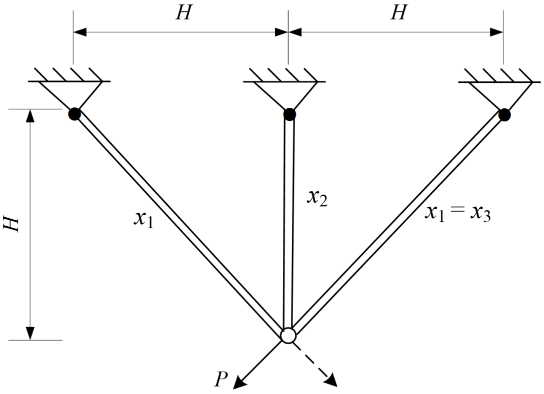

Figure 9. The structure of three-bar truss design problem.

The mathematical form of this problem is given as follows: Minimize $f(X) = (2\sqrt{2}x_1 + x_2) \times l$, $X = [x_1, x_2]$

$$
\begin{aligned}
g_1(X) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\
g_2(X) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\
g_3(X) &= \frac{1}{x_1 + \sqrt{2}x_2}P - \sigma \leq 0
\end{aligned}
$$

where $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $l = 100\,\text{cm}$, $P = 2\,\text{KN/cm}^2$, and $\sigma = 2\,\text{KN/cm}^2$.

The three-bar truss problem is solved by the EGWO algorithm and the result is given in Table 7. In EGWO, the population size is 50 and the maximum number of iteration is 5000. Additionally, the results obtained by the other nine optimization methods such as CMAES, JADE, WPSO, GABC [57], GWO, mGWO, EEGWO, IGWO, and MGWO are also provided in Table 7.

As shown in Table 7, compared to the other nine optimization techniques, the proposed EGWO algorithm obtains the minimum volume for three-bar truss design problem.

4.2.3 Pressure vessel design problem

As shown in Figure 10, this problem considers a pressure vessel structure. The purpose of this case is to obtain the minimum cost of a pressure vessel with four inequality constraints. This problem has four decision variables namely thickness of the shell $T_s(x_1)$ and the head $T_h(x_2)$, inner radius $R(x_3)$ and the length of the cylindrical section $L(x_4)$.

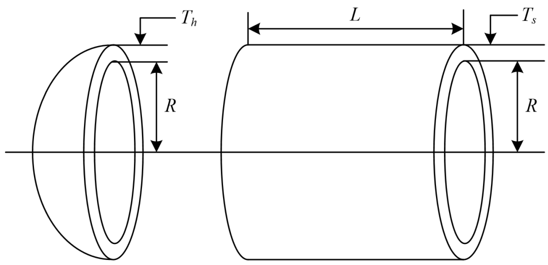

Figure 10. The structure of pressure vessel design problem.

Mathematically, this problem is formulated by:

Minimize $f(\bar{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$, $X = [x_1, x_2, x_3, x_4]$

$$
\begin{aligned}
g_1(\bar{x}) &= -x_1 + 0.0193x_3 \leq 0 \\
g_2(\bar{x}) &= -x_2 + 0.00954x_3 \leq 0 \\
g_3(\bar{x}) &= -\pi x_3^2 x_4 + \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
g_4(\bar{x}) &= x_4 - 240 \leq 0
\end{aligned}
$$

where $1 \leq x_1, x_2 \leq 99$ and $10 \leq x_3, x_4 \leq 200$.

Applying the EGWO to solve the pressure vessel design problem and its result is outlined in Table 8. The population size of EGWO is 50 and the maximum number of iteration is 5000. In addition, the results of several GWO variants and other meta-heuristic algorithms are also given in Table 8.

From Table 8, CMAES and MGWO obtain the optimal objective function value (6059.714) for pressure vessel design problem. Compared to the CPSO, GABC, WOA, GWO, mGWO, EEGWO, and IGWO algorithms, EGWO gets the better objective function value. However, the better results are obtained by the CMAES, JADE, WPSO, MGWO and ISCA algorithms.

4.3 EGWO for feature selection problems

Feature selection (FS) is a crucial pre-processing step in the classification, regression, forecasting and data

Table 6. The results obtained by EGWO and other approaches for spring problem.

Algorithm	Optimal values for variables			Objective value
	$d(x_1)$	$D(x_2)$	$P(x_3)$	
CMAES	0.051129	0.343403	12.11483	0.012677
JADE	0.051129	0.343403	12.11483	0.012677
CPSO	0.051728	0.357644	11.244543	0.0126747
GWO	0.051129	0.343403	12.11483	0.012677
mGWO	0.051445	0.350456	11.69129	0.012699
EEGWO	0.051665	0.355896	11.40440	0.012734
IGWO	0.051129	0.343403	12.11483	0.012677
MGWO	0.051640	0.355530	11.36064	0.012668
ISCA	0.0520217	0.364768	10.83230	0.012667
EGWO (this study)	0.0515088	0.352393	11.10210	0.012667

Table 7. The results obtained by EGWO and other approaches for three-bar truss problem.

Algorithm	Optimal values for variables		Objective function value
	x_1	x_2	
CMAES	0.78623	0.41608	263.9867
JADE	0.78753	0.41150	263.8968
WPSO	0.58959	0.20568	263.8994
GABC	0.78784	0.41062	263.8966
GWO	0.7898013	0.4050788	263.8974
mGWO	0.7878452	0.4106108	263.8974
EEGWO	0.8087517	0.3592847	264.6780
IGWO	0.7878452	0.4106108	263.8974
MGWO	0.7885845	0.4085071	263.8971
EGWO (this study)	0.78852	0.40869	263.8959

mining applications. The purpose of FS is for choosing the most significant features from the original features to reduce the dimensionality of the datasets. However, choosing the best feature subsets from the original datasets are the very challenging issues, especially for datasets with high dimensionality [5]. In essence, FS is a typical combinatorial optimization problem which requires lots of computation. During the past few years, many meta-heuristic optimization techniques are suggested for dealing with the FS problems [58–61].

In this section, EGWO is applied to tackle the FS problem for verifying its search performance. However, the solution space of FS is represented by binary values. It must be noted that EGWO is a continuous-space-based optimization method which needs to transform it from continuous version into binary one when solving the FS problems. One of the easiest conversion techniques is to use a transfer function. The advantage of this technique is not to modify the framework of the GWO algorithm. This paper uses the S-shaped transfer function as follows:

$$T(x) = \frac{1}{1+e^{-rx}} \tag{19}$$

where r is a constant number.

To investigate the overall performance of EGWO, sixteen representative datasets from the UCI data repository are used for experiments [58]. The detailed information of these datasets such as the number of features and the number of samples are given in Table 9.

For all the datasets, the wrapper approach on feature subsets selection is used with KNN classifier ($k = 5$). In EGWO, the population number is 10, the maximum iterative number is 100, the number of run is 20, and the dimensions of problem are equal to the number of features for each dataset. The overall performance of EGWO is compared against ant lion optimizer (ALO) algorithm [62], BOA [58], crow search algorithm (CSA) [63], PSO [64], GWO [3], Augmented GWO

Table 8. Comparison between EGWO and other approaches for pressure vessel problem.

Algorithm	Optimal values for variables				Objective value
	$T_s(x_1)$	$T_h(x_2)$	$R(x_3)$	$L(x_4)$	
CMAES	0.8125	0.4375	42.09844	176.6366	6059.7140
JADE	0.8125	0.4375	42.09844	176.6366	6059.7144
WPSO	0.8125	0.4375	42.09844	176.6365	6059.7143
CPSO	0.8125	0.4375	42.0913	176.7465	6061.0777
GABC	1.0625	0.5625	54.7796	68.0057	6934.8979
WOA	12.96996	7.337754	42.03656	177.4064	6067.2991
GWO	13.18886	7.349468	42.09791	176.6464	6059.8523
mGWO	12.99563	6.865056	42.09786	176.6465	6059.8490
EEGWO	13.09291	6.792196	42.09758	176.6495	6059.8704
IGWO	12.85317	6.980472	42.09806	176.6416	6059.9066
MGWO	0.8125	0.4375	42.09844	176.6366	6059.7140
ISCA	12.96419	7.150134	42.09829	176.6392	6059.7489
EGWO (this study)	12.81254	7.031335	42.09819	176.6401	6059.7518

Table 9. The detailed characteristics of sixteen datasets used in the experiments.

Number	Dataset	Number of features	Number of samples
1	Breastcancer	9	699
2	BreastEW	30	569
3	CongressEW	16	435
4	Exactly	13	1000
5	Exactly2	13	1000
6	HeartEW	13	270
7	IonosphereEW	34	351
8	Lymphography	18	148
9	M-of-n	13	1000
10	PenglungEW	325	73
11	SonarEW	60	208
12	SpectEW	22	267
13	Tic-tac-toe	9	958
14	Vote	16	300
15	WineEW	13	178
16	Zoo	16	101

(AGWO) [65], and enhanced GWO (eGWO) [66]. Table 10 provides the classification accuracy of EGWO and other compared approaches. Table 11 outlines the optimal feature subset of EGWO against other algorithms. It should be noted that the results of ALO, BOA, and GWO are from Arora et al. [58], and the results of CSA, PSO, eGWO, and AGWO are taken from Arora, et al. [29].

From Table 10, the classification accuracy of EGWO is significantly superior to other compared approaches on all the datasets. Based on the results shown in Table 11, compared with ALO, BOA, PSO, GWO, AGWO, and eGWO, EGWO gets the fewer number of selected features on all the datasets. In addition, EGWO performs better than CSA on all the datasets except for BreastEW dataset. Figure 11 shows the mean ranking of the classification accuracy and number of selected features on 16 datasets for eight algorithms.

As seen in Figure 11, for classification accuracy, the EGWO obtains the first rank, followed by the BOA, AGWO, eGWO, CSA, PSO, ALO, and GWO. Additionally, the EGWO ranks the first, followed by the CSA, PSO, AGWO, GWO, eGWO, BOA, and ALO for the number of selected features.

5 Discussion

The results demonstrate that the proposed EGWO consistently outperforms the basic GWO and other seven state-of-the-art algorithms on most benchmark functions. To better understand the source of this improvement, the contributions of the three embedded strategies were separately investigated. The nonlinear transition mechanism significantly enhances the balance between exploration and exploitation, accelerating convergence at later iterations. The cuckoo search component improves local search capability by introducing Lévy flight behavior, which helps EGWO escape from local optima. The adaptive position search equation contributes to maintaining population diversity and prevents premature convergence. Empirically, the combination of the nonlinear transition rule and adaptive position search equation accounts for most of the observed performance gain, while the cuckoo search strategy further refines

Table 10. Comparison between EGWO and other algorithms for classification accuracy.

Datasets	ALO	BOA	CSA	PSO	GWO	AGWO	eGWO	EGWO
Breastcancer	0.9591	0.9640	0.9606	0.9609	0.9603	0.9606	0.9617	0.9705
BreastEW	0.9392	0.9506	0.9460	0.9378	0.9375	0.9340	0.9474	0.9556
CongressEW	0.9370	0.9459	0.9330	0.9235	0.9327	0.9358	0.9431	0.9521
Exactly	0.7061	0.7676	0.7664	0.7471	0.7249	0.7576	0.7536	0.8054
Exactly2	0.6980	0.7189	0.6908	0.6959	0.6929	0.6956	0.6988	0.7294
HeartEW	0.7773	0.8010	0.7822	0.7788	0.7768	0.7970	0.7615	0.8288
IonosphereEW	0.8595	0.9023	0.8943	0.8803	0.8682	0.8932	0.8636	0.9042
Lymphography	0.7863	0.8224	0.7919	0.7906	0.7629	0.7919	0.7663	0.8797
M-of-n	0.8184	0.8511	0.8560	0.8425	0.8272	0.8780	0.8704	0.9154
PenglungEW	0.8072	0.8665	0.8054	0.8140	0.8341	0.8541	0.7568	0.8832
SonarEW	0.8487	0.8853	0.8538	0.8667	0.8622	0.8827	0.8615	0.8870
SpectEW	0.7881	0.8154	0.7925	0.7841	0.7846	0.8134	0.8045	0.8662
Tic-tac-toe	0.7587	0.7744	0.7649	0.7518	0.7537	0.7628	0.7712	0.7990
Vote	0.9258	0.9369	0.9213	0.9258	0.9196	0.9200	0.9027	0.9590
WineEW	0.9543	0.9715	0.9618	0.9521	0.9476	0.9573	0.9663	0.9839
Zoo	0.9216	0.9608	0.9373	0.9451	0.9525	0.9686	0.9686	0.9742
Average	0.8428	0.8709	0.8536	0.8498	0.8461	0.8627	0.8499	0.9167

Table 11. Number of selected features for EGWO and other algorithms.

Datasets	ALO	BOA	CSA	PSO	GWO	AGWO	eGWO	EGWO
Breastcancer	7	6.2	6.4	5.7	6.9	5.2	6.6	**4.8**
BreastEW	24.27	21.07	**14.4**	18.33	19	19.2	20.4	15.6
CongressEW	9.87	9.4	8.8	10.8	9.8	9.8	11.2	**5.2**
Exactly	12.87	10.87	8.6	9	12.07	10.8	9.2	6
Exactly2	8.4	7.87	6.8	9.4	7.53	7.4	9.8	**2.6**
HeartEW	10.4	9.13	8.4	9.07	8.8	8.6	8.2	**7.2**
IonosphereEW	20.13	22.27	15.6	19.2	17.33	22.6	21	**9.4**
Lymphography	13.33	13.27	10	11.73	11.8	10.4	11.6	**9.4**
M-of-n	11.27	11.07	8.4	10.87	11.27	10.2	11.2	6
PenglungEW	172.07	170.8	157.8	183.33	162.8	158.6	177.4	**149.6**
SonarEW	48	41.07	30.2	37.6	41.6	47	41.6	**27**
SpectEW	13.87	15.2	11.8	12.07	13.2	17.8	12	**8.8**
Tic-tac-toe	8.8	7.87	6.6	6.73	7.53	7.8	7.6	**6.6**
Vote	8.4	9.6	6.8	9.33	8.47	9.2	8.8	**5.4**
WineEW	11.07	10.27	8	10.07	10.73	10.6	9.6	**6.4**
Zoo	11.67	10.47	8.2	11.8	12.4	11.2	12.8	**7**
Average	24.46	23.53	19.8	23.44	22.58	22.9	23.69	**17.31**

exploitation efficiency. This analysis suggests that a synergistic integration of exploration-oriented and exploitation-oriented mechanisms is the key to EGWO's superior performance.

6 Conclusions

This work developed an enhanced grey wolf optimizer with multiple strategies, named as EGWO, to deal with various numerical optimization problems. In EGWO, a good transition balance between diversity and convergence was achieved by designing a new nonlinear transition parameter. Secondly, an adaptive position search equation based on a dynamic parameter was proposed to effectively coordinate the global and local search capability. Finally, the cuckoo search algorithm was used for the decision layer individuals to improve the exploitation ability of GWO. The effectiveness of EGWO was studied on 25 classical benchmark functions. Moreover, three engineering optimization design problems and the feature selection problems with 16 datasets were applied to verify the feasibility of EGWO. The

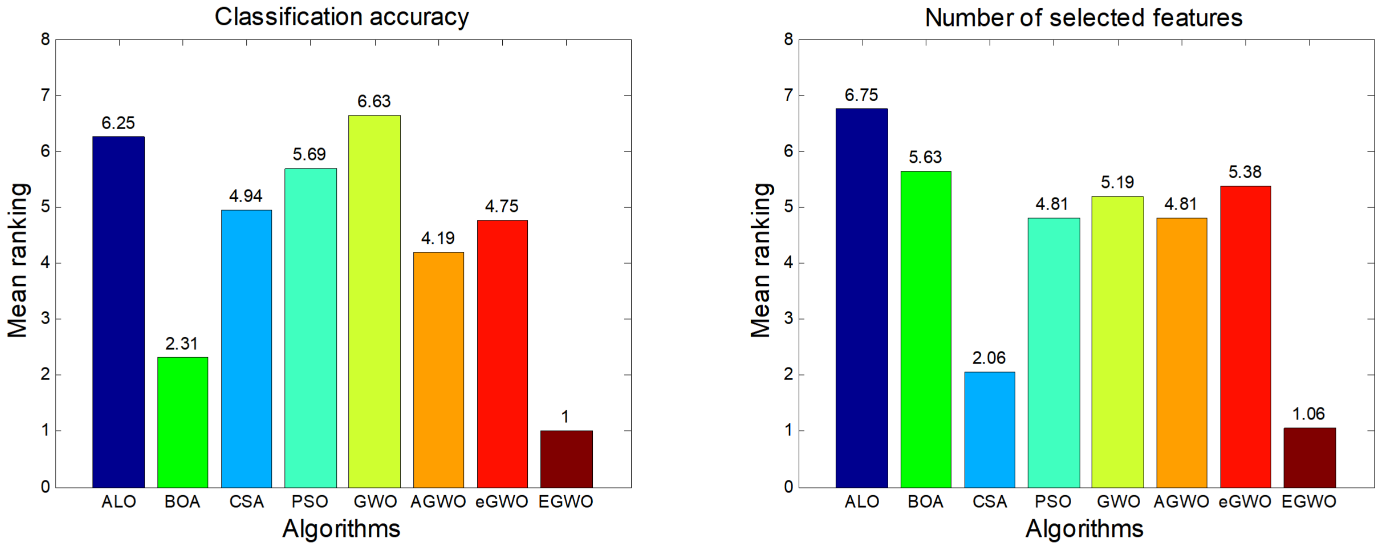

Figure 11. Mean Friedman test ranks of eight algorithms on 16 benchmark datasets.

comparison results indicated that EGWO not only enhanced the solution accuracy, but also accelerated the convergence speed on benchmark functions as well as real-world optimization problems.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 12361106; in part by the Guizhou Provincial Science and Technology Plan Key Projects of Qiankehe Jichu under Grant ZK[2023]003; in part by the Guizhou Provincial High Level Innovative Talent Training Plan Project of Qiankehe Platform Talent under Grant GCC[2023]006.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Gharehchopogh, F. S., Nadimi-Shahraki, M. H., Barshandeh, S., Abdollahzadeh, B., & Zamani, H. (2023). Cqffa: A chaotic quasi-oppositional farmland fertility algorithm for solving engineering optimization problems. *Journal of Bionic Engineering, 20*(1), 158-183. [CrossRef]

[2] Abualigah, L., Diabat, A., Svetinovic, D., & Elaziz, M. A. (2023). Boosted Harris Hawks gravitational force algorithm for global optimization and industrial engineering problems. *Journal of Intelligent Manufacturing, 34*(6), 2693-2728. [CrossRef]

[3] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. [CrossRef]

[4] Nadimi-Shahraki, M. H., Moeini, E., Taghian, S., & Mirjalili, S. (2023). Discrete improved grey wolf optimizer for community detection. *Journal of Bionic Engineering, 20*(5), 2331-2358. [CrossRef]

[5] Long, W., Cai, S., Jiao, J., Xu, M., & Wu, T. (2020). A hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. *Energy Conversion and Management, 203*, 112243. [CrossRef]

[6] Nuaekaew, K., Artrit, P., Pholdee, N., & Bureerat, S. (2017). Optimal reactive power dispatch problem using a two-archive multi-objective grey wolf optimizer. *Expert Systems with Applications, 87*, 79–89. [CrossRef]

[7] Makhadmeh, S. N., Khader, A. T., Al-Betar, M. A., Naim, S., Abasi, A. K., & Alyasseri, Z. A. A. (2021). A hybrid grey wolf optimizer with min-conflict algorithm for power scheduling in a smart home. *Swarm and Evolutionary Computation, 60*, 100793. [CrossRef]

[8] Ahmadi, R., Ekbatanifard, G., & Bayat, P. (2021). A modified grey wolf optimizer-based data clustering algorithm. *Applied Artificial Intelligence, 35*(1), 63–79. [CrossRef]

[9] Nadimi-Shahraki, M. H., Taghian, S., & Mirjalili, S. (2021). An improved grey wolf optimizer for solving engineering problems. *Expert Systems with Applications, 166*, 113917. [CrossRef]

[10] Altan, A., Karasu, S., & Zio, E. (2021). A hybrid model for wind speed forecasting combining LSTM neural network, decomposition methods and grey

wolf optimizer. *Applied Soft Computing*, *100*, 106996. [CrossRef]

[11] Yu, X., & Wu, X. (2022). Ensemble grey wolf optimizer and its application for image segmentation. *Expert Systems with Applications*, *209*, 118267. [CrossRef]

[12] Abualigah, L., Habash, M., Hanandeh, E. S., Hussein, A. M., Shinwan, M. A., Zitar, R. A., & Jia, H. (2023). Improved reptile search algorithm by salp swarm algorithm for medical image segmentation. *Journal of bionic engineering*, *20*(4), 1766-1790. [CrossRef]

[13] Jia, Y., Qu, L., & Li, X. (2023). Automatic path planning of unmanned combat aerial vehicle based on double-layer coding method with enhanced grey wolf optimizer. *Artificial Intelligence Review, 56*(10), 12257-12314. [CrossRef]

[14] Manikandan, K. (2019). Diagnosis of diabetes diseases using optimized fuzzy rule set by grey wolf optimization. *Pattern Recognition Letters*, *125*, 432-438. [CrossRef]

[15] Singh, D., & Dhillon, J. S. (2019). Ameliorated grey wolf optimizer for economic load dispatch problem. *Energy*, *169*, 398–419. [CrossRef]

[16] Srikanth, K., Panwar, L. K., Panigrahi, B. K., Herrera-Viedma, E., Sangaiah, A. K., & Wang, G. (2018). Quantum-inspired binary grey wolf optimizer for unit commitment problem. *Computers & Electrical Engineering*, *70*, 243–260. [CrossRef]

[17] Ebrahimi, S. M., Hasanzadeh, S., & Khatibi, S. (2023). Parameter identification of fuel cell using repairable grey wolf optimization algorithm. *Applied Soft Computing*, *147*, 110791. [CrossRef]

[18] Gharehchopogh, F. S. (2022). An improved tunicate swarm algorithm with best-random mutation strategy for global optimization problems. *Journal of Bionic Engineering*, *19*(4), 1177-1202. [CrossRef]

[19] Wang, Z., Shu, L., Yang, S., Zeng, Z., He, D., & Chan, S. (2025). Multi-population dynamic grey wolf optimizer based on dimension learning and Laplace Mutation for global optimization. *Expert Systems with Applications*, *265*, 125863. [CrossRef]

[20] Heidari, A. A., & Pahlavani, P. (2017). Modified grey wolf optimizer with Lévy flight for optimization tasks. *Applied Soft Computing*, *60*, 115–134. [CrossRef]

[21] Hu, P., Pan, J. S., & Chu, S. C. (2020). Improved binary grey wolf optimizer and its application for feature selection. *Knowledge-Based Systems*, *195*, 105746. [CrossRef]

[22] Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. *Applied computational intelligence and soft computing*, *2016*(1), 7950348. [CrossRef]

[23] Singh, N. (2020). A modified variant of grey wolf optimizer. *Scientia Iranica*, *27*(3), 1450-1466. [CrossRef]

[24] Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C. I., & Soto, J. (2017). Fuzzy hierarchical operator in grey wolf optimizer. *Applied Soft Computing*, *57*, 315–328. [CrossRef]

[25] Daniel, E., Anitha, J., & Gnanaraj, J. (2017). Optimum laplacian wavelet mask based medical image using hybrid cuckoo search–grey wolf optimization algorithm. *Knowledge-Based Systems*, *131*, 58-69. [CrossRef]

[26] Yu, X., Zhao, Q., Lin, Q., & Wang, T. (2023). A grey wolf optimizer-based chaotic gravitational search algorithm for global optimization. *The Journal of Supercomputing*, *79*(3), 2691-2739. [CrossRef]

[27] Sundaramurthy, S., & Jayavel, P. (2020). A hybrid grey wolf optimization and particle swarm optimization with C4. 5 approach for prediction of rheumatoid arthritis. *Applied Soft Computing*, *94*, 106500. [CrossRef]

[28] Singh, G., & Singh, U. (2023). Hybrid binary grey wolf naked mole-rat algorithm for fragment-type UWB antenna optimization using time-varying transfer functions. *Expert Systems with Applications*, *232*, 120802. [CrossRef]

[29] Arora, S., Singh, H., Sharma, M., Sharma, S., & Anand, P. (2019). A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection. *IEEE Access*, *7*, 26343-26361. [CrossRef]

[30] Zhou, J., Huang, S., Zhou, T., Armaghani, D. J., & Qiu, Y. (2022). Employing a genetic algorithm and grey wolf optimizer for optimizing RF models to evaluate soil liquefaction potential. *Artificial intelligence review*, *55*(7), 5673-5705. [CrossRef]

[31] Shilaja, C., & Arunprasath, T. (2019). Internet of medical things-load optimization of power flow based on hybrid enhanced grey wolf optimization and dragonfly algorithm. *Future Generation Computer Systems*, *98*, 319-330. [CrossRef]

[32] Hemalatha, S., Banu, G., & Indirajith, K. (2023). Design and investigation of PV string/central architecture for bayesian fusion technique using grey wolf optimization and flower pollination optimized algorithm. *Energy Conversion and Management*, *286*, 117078. [CrossRef]

[33] Yang, Q., Liu, J., Wu, Z., & He, S. (2023). A fusion algorithm based on whale and grey wolf optimization algorithm for solving real-world optimization problems. *Applied Soft Computing*, *146*, 110701. [CrossRef]

[34] Karakoyun, M., Ozkis, A., & Kodaz, H. (2020). A new algorithm based on gray wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems. *Applied Soft Computing*, *96*, 106560. [CrossRef]

[35] Makhadmeh, S. N., Abasi, A. K., & Al-Betar, M. A. (2022). Hybrid multi-verse optimizer with grey wolf optimizer for power scheduling problem in smart

home using IoT. *The Journal of Supercomputing*, *78*(9), 11794-11829. [CrossRef]

[36] Dhargupta, S., Ghosh, M., Mirjalili, S., & Sarkar, R. (2020). Selective opposition-based grey wolf optimizer. *Expert Systems with Applications*, *151*, 113389. [CrossRef]

[37] Khanum, R. A., Jan, M. A., Aldegheishem, A., Mehmood, A., Alrajeh, N., & Khanan, A. (2019). Two new improved variants of grey wolf optimizer for unconstrained optimization. *IEEE Access*, *8*, 30805-30825. [CrossRef]

[38] Gupta, S., & Deep, K. (2019). A novel random walk grey wolf optimizer. *Swarm and evolutionary computation*, *44*, 101-112. [CrossRef]

[39] Abdel-Basset, M., El-Shahat, D., El-Henawy, I., De Albuquerque, V. H. C., & Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, *139*, 112824. [CrossRef]

[40] Ahmed, R., Rangaiah, G. P., Mahadzir, S., Mirjalili, S., Hassan, M. H., & Kamel, S. (2023). Memory, evolutionary operator, and local search based improved Grey Wolf Optimizer with linear population size reduction technique. *Knowledge-Based Systems*, *264*, 110297. [CrossRef]

[41] Kumar, V., & Kumar, D. (2017). An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems. *Advances in Engineering Software*, *112*, 231-254. [CrossRef]

[42] Rodríguez, A., Camarena, O., Cuevas, E., Aranguren, I., Valdivia-G, A., Morales-Castañeda, B., ... & Pérez-Cisneros, M. (2021). Group-based synchronous-asynchronous grey wolf optimizer. *Applied Mathematical Modelling*, *93*, 226-243. [CrossRef]

[43] Seyyedabbasi, A., & Kiani, F. (2021). I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. *Engineering with Computers*, *37*(1), 509-532. [CrossRef]

[44] Sang-To, T., Le-Minh, H., Mirjalili, S., Wahab, M. A., & Cuong-Le, T. (2022). A new movement strategy of grey wolf optimizer for optimization problems and structural damage identification. *Advances in Engineering Software*, *173*, 103276. [CrossRef]

[45] Yang, X. S., & Deb, S. (2010). Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, *1*(4), 330-343. [CrossRef]

[46] Yang, X. S., & Deb, S. (2013). Multiobjective cuckoo search for design optimization. *Computers & Operations Research*, *40*(6), 1616-1624. [CrossRef]

[47] Nickabadi, A., Ebadzadeh, M. M., & Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied soft computing*, *11*(4), 3658-3670. [CrossRef]

[48] Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, *9*(2), 159-195. [CrossRef]

[49] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition-based differential evolution. *IEEE Transactions on Evolutionary computation*, *12*(1), 64-79. [CrossRef]

[50] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, *95*, 51-67. [CrossRef]

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

[52] Lu, C., Gao, L., Li, X., Hu, C., Yan, X., & Gong, W. (2020). Chaotic-based grey wolf optimizer for numerical and engineering optimization problems. *Memetic Computing*, *12*(4), 371–398. [CrossRef]

[53] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, *186*(2-4), 311-338. [CrossRef]

[54] Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, *13*(5), 945-958. [CrossRef]

[55] He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering applications of artificial intelligence*, *20*(1), 89-99. [CrossRef]

[56] Long, W., Wu, T., Liang, X., & Xu, S. (2019). Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert systems with applications*, *123*, 108-126. [CrossRef]

[57] Gao, W., Liu, S., & Huang, L. (2012). A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics*, *236*(11), 2741-2753. [CrossRef]

[58] Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, *116*, 147-160. [CrossRef]

[59] Gharehchopogh, F. S. (2023). An improved Harris Hawks optimization algorithm with multi-strategy for community detection in social network. *Journal of Bionic Engineering*, *20*(3), 1175-1197. [CrossRef]

[60] Long, W., Jiao, J., Liang, X., Wu, T., Xu, M., & Cai, S. (2021). Pinhole-imaging-based learning butterfly optimization algorithm for global optimization and feature selection. *Applied Soft Computing*, *103*, 107146. [CrossRef]

[61] Abualigah, L., & Dulaimi, A. J. (2021). A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm. *Cluster Computing*, *24*(3), 2161-2176. [CrossRef]

[62] Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary ant lion approaches for feature selection. *Neurocomputing, 213*, 54-65. [CrossRef]

[63] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & structures, 169*, 1-12. [CrossRef]

[64] Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In 1998 IEEE international conference on evolutionary computation proceedings. *IEEE world congress on computational intelligence* (*Cat. No. 98TH8360*) (pp. 69-73). IEEE. [CrossRef]

[65] Qais, M. H., Hasanien, H. M., & Alghuwainem, S. (2018). Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems. *Applied Soft Computing, 69*, 504-515. [CrossRef]

[66] Joshi, H., & Arora, S. (2017). Enhanced grey wolf optimization algorithm for global optimization. *Fundamenta Informaticae, 153*(3), 235-264. [CrossRef]

Wenmin Xu is currently pursuing a Master's degree in Computational Mathematics at Guizhou University of Finance and Economics. (Email: 1907891682@qq.com)

Wen Long received the M.S. degree in system science from Guangxi Normal University, Guilin, China, in 2008, and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2011. (Email: lw227@mail.gufe.edu.cn)