Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085316

ARTICLE

Weighted Fuzzy Production Rule Extraction Utilizing an Improved Grey Wolf Optimizer

Xue-Wei Liu[1], Shao-Qiang Ye[2], Feng Qin[3] and Kai-Qing Zhou[1,*]

[1]College of Computer Science and Engineering, Jishou University, Jishou, China
[2]Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor Bahru, Malaysia
[3]School of Computer and Artificial Intelligence, Huaihua University, Huaihua, China

*Corresponding Author: Kai-Qing Zhou. Email: kqzhou@jsu.edu.cn

Received: 08 May 2026; Accepted: 14 July 2026; Published: 13 August 2026

ABSTRACT: Weighted fuzzy production rules (WFPRs) provide superior expressiveness and interpretability in knowledge engineering area. However, manual construction of WFPRs is labor-intensive, time-consuming, and inherently subjective, which greatly restricts their practical application. The back propagation neural network (BPNN) has been widely adopted for automatic WFPR extraction. Nevertheless, its high sensitivity to initial weight configurations frequently results in premature convergence to local optima, generating redundant, poorly interpretable rule sets that compromise the inherent interpretability advantage of WFPRs. This paper proposes an elite dynamic scout-guided grey wolf optimizer (EDSG-GWO) and integrates it into a BPNN-based WFPR extraction framework to optimize network initial weights. The EDSG-GWO incorporates a nonlinear convergence factor, dynamic weighted position updating, an elite opposition-based learning mechanism, and an adaptive scout bee perturbation strategy to effectively balance global exploration and local exploitation. Unlike existing GWO variants, the EDSG-GWO achieves the synergistic integration of the four above strategies, collectively enhancing convergence accuracy and exploration capability. Numerical experiments are conducted on twelve benchmark functions. The results demonstrate that the EDSG-GWO delivers competitive optimization accuracy and convergence speed. Validated on the PIMA Indians Diabetes Database, the optimized BPNN attains a test accuracy of 72.92%, which is comparable to other metaheuristic-based approaches. More notably, the extracted WFPRs reach an accuracy of 77.08%, outperforming the baseline method by a notable margin. The four extracted rules involve merely six core diagnostic features, whose weight distributions are highly consistent with established medical knowledge. This contributes to a concise, clinically plausible, and highly interpretable rule set for auxiliary diabetes diagnosis. Further validation on the Breast Cancer Wisconsin dataset yields a WFPRs testing accuracy of 94.15%, confirming the framework's generalizability.

KEYWORDS: Grey wolf optimizer; function optimization; back propagation neural network; weighted fuzzy production rule extraction

1 Introduction

Expert systems (ESs) emulate human expert reasoning through a knowledge base and inference engine, with their effectiveness largely depending on knowledge representation [1]. Traditional ESs rely on deterministic rules inadequate for handling real-world uncertainty. To address this, fuzzy logic represents uncertain knowledge through fuzzy production rules using continuous membership degrees, and weighted fuzzy production rules (WFPRs) further assign different weights to premises to reflect the relative importance

of features. Due to their ability to manage uncertainty, fuzzy expert systems have been widely applied in domains such as medical diagnosis [2].

Automatically acquiring reusable and interpretable knowledge from data remains a core challenge in constructing knowledge-based intelligent systems [3,4]. In fuzzy expert systems, this challenge manifests as how to automatically extract accurate and concise weighted fuzzy production rules from data [5]. The back propagation neural network (BPNN) has been widely adopted for WFPR extraction owing to its strong nonlinear mapping capability [6]; however, it is highly sensitive to initial weights, and poor initialization may cause unstable predictions and redundant rule generation, reducing interpretability. Consequently, metaheuristic algorithms have increasingly been employed to optimize the initial weights of BPNN [7].

Among them, grey wolf optimizer (GWO), inspired by the social hierarchy and cooperative hunting behavior of grey wolves, has attracted considerable attention due to its simple structure, few control parameters, and effectiveness in solving diverse optimization problems [8]. However, GWO still suffers from slow convergence, insufficient population diversity, and premature convergence. To address these shortcomings, various improvement strategies have been proposed. Li et al. [9] integrated adaptive weighting and Lévy flight perturbation into the sparrow search algorithm to balance local and global search. Zhang and Cai [10] proposed employing chaotic mapping and adaptive self-learning strategies to enhance population diversity and broaden global coverage. These studies demonstrate that multi-strategy enhancements can effectively improve the search performance of swarm intelligence algorithms.

To address these limitations, this paper proposes an elite dynamic scout-guided GWO (EDSG-GWO) to enhance both exploration and exploitation. The main contributions are summarized as follows:

(i) An improved GWO algorithm integrating a nonlinear convergence factor, dynamic weighted position update, elite opposite-based learning, and adaptive perturbation strategy to improve convergence accuracy, global search capability, and escape from local optima.
(ii) An EDSG-GWO–BPNN hybrid framework for optimizing the initial weights of BPNN, thereby improving training efficiency and classification accuracy.
(iii) An interpretable WFPR extraction method that transforms the black-box BPNN into a transparent rule-based system for constructing fuzzy expert systems with both high interpretability and predictive performance.

The remainder of this paper is organized as follows. Section 2 presents the proposed framework and algorithm. Section 3 introduces the data preprocessing, BPNN training, and WFPR extraction procedures. Section 4 provides experimental validation and result analysis. Finally, Section 5 concludes the paper and discusses future research directions.

2 Methods

This section introduces the standard GWO and its improved version, EDSG-GWO. First, the basic principles and working mechanisms of GWO are presented. Then, the four improvement strategies of EDSG-GWO are discussed in detail, namely the nonlinear convergence factor, the dynamic weighted position update strategy, the elite opposite-based learning mechanism, and the adaptive scout bee perturbation strategy. The resulting optimizer will subsequently be employed in Section 4.2 to optimize the BPNN weights for weighted fuzzy production rule extraction.

2.1 Standard Grey Wolf Optimizer

GWO simulates the social hierarchy of grey wolves, in which the population is sorted by fitness and the top three individuals are designated α, β and δ, while the remainder are ω. Position updates are guided by

these three leaders via Eqs. (1)–(3):

$$\begin{cases} \vec{D}_\alpha = \left| \vec{C}_1 \otimes \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta = \left| \vec{C}_2 \otimes \vec{X}_\beta - \vec{X} \right| \\ \vec{D}_\delta = \left| \vec{C}_3 \otimes \vec{X}_\delta - \vec{X} \right| \end{cases} \tag{1}$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \otimes \vec{D}_\alpha \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \otimes \vec{D}_\beta \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \otimes \vec{D}_\delta \end{cases} \tag{2}$$

$$\vec{X}_{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{3}$$

The above equation, t is the iteration index, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_\alpha$, $\vec{X}_\beta$, $\vec{X}_\delta$ are the leader positions, $\vec{X}$ is an individual wolf's position, and $\otimes$ denotes element-wise multiplication. The coefficient vectors are

$$\vec{A} = 2\vec{a} \otimes \vec{r}_1 - \vec{a} \tag{4}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{5}$$

where a decreases linearly from 2 to 0 over the iterations and $\vec{r}_1$, $\vec{r}_2$ are uniform random vectors in $[0, 1]$. Eq. (1) computes the distance to each leader, Eq. (2) derives candidate positions, and Eq. (3) averages the three candidates to obtain the updated position. GWO is simple and efficient for low-dimensional problems but tends to stagnate in local optima when dimensionality grows.

2.2 Elite Dynamic Scout-Guided Grey Wolf Optimizer

Standard GWO relies on three leader wolves to steer the population toward promising regions, yet its fixed linear decay of the convergence factor and uniform leader weighting frequently cause the swarm to stagnate in local optima, especially as dimensionality grows. EDSG-GWO addresses these deficiencies through four coordinated improvement strategies, each targeting a distinct aspect of the search dynamics.

2.2.1 Nonlinear Convergence Factor

In standard GWO, a drops linearly from 2 to 0 without adapting to search progress. To prolong early exploration and accelerate late refinement, a is redefined as a quadratic function of the iteration ratio:

$$a = 2\left[1 - \left(\frac{t}{T}\right)^2\right] \tag{6}$$

In this formulation, t is the elapsed iteration count and T is the maximum. The quadratic decay keeps a large early on, preserving exploration, and steepens later, accelerating exploitation.

2.2.2 Dynamic Weighted Position Update

In standard GWO, α, β and δ wolves are treated equally, with each contributing one third to every position update. This uniform scheme overlooks the reality that α almost always occupies a far better region than δ. EDSG-GWO therefore assigns coefficients that depend on the iteration and progressively amplify α's influence. The three leader weights are defined in Eq. (7):

$$\begin{cases} w_\alpha = 0.4 + 0.3r \\ w_\beta = 0.35 - 0.15r \\ w_\delta = 0.25 - 0.15r \end{cases} \tag{7}$$

with $r = \frac{t}{T}$ denoting normalized progress. The three coefficients always sum to unity. Early in the search, the weights are nearly balanced, preserving exploratory breadth; as iterations accumulate, α share grows, guiding the population more strongly toward the best-known region.

In addition, a dynamic contraction coefficient is introduced to regulate the magnitude of the updated position. The weighted blend of the three leaders is first computed as in Eq. (8):

$$\vec{x}^{*} = w_1 X_1 + w_2 X_2 + w_3 X_3 \tag{8}$$

The contraction coefficient is then defined as Eq. (9):

$$\lambda(t) = 1.0 - 0.5 \cdot \left(\frac{t}{T}\right)^2 \tag{9}$$

Finally, the updated position of each wolf is obtained as shown in Eq. (10):

$$\vec{X}_{\text{new}} = \lambda(t) \cdot \vec{x}^{*} \tag{10}$$

The coefficient starts at 1.0 in the first iteration, preserving the standard weighted-update magnitude, and gradually decreases to 0.5, thereby pulling updated positions toward a progressively contracted region in the later stage. Combined with the nonlinear convergence factor in Eq. (6), it provides a smooth transition from global exploration to local exploitation.

2.2.3 Elite Opposite-Based Learning

Population diversity inevitably erodes as individuals cluster around the leaders. One countermeasure is OBL, which generates a symmetric solution within the current search boundaries and keeps the better one. However, applying OBL to every wolf doubles the number of function evaluations per iteration. EDSG-GWO limits the overhead by selecting only the top 10% of individuals ranked by fitness. These individuals are the closest to the currently known optima and are thus most likely to generate high-quality opposite solutions. Their opposites are then generated within dynamically updated bounds. Let da_d and db_d be the current minimum and maximum in each dimension across the population. The opposite of an elite position x_i in dimension d is computed by Eq. (11):

$$\tilde{x}_{i,d} = k_d \cdot (da_d + db_d - x_{i,d}), \quad d = 1, 2, \ldots, D \tag{11}$$

In Eq. (11), $k_d \in [0,1]$ is a uniform random coefficient in each dimension that introduces controlled randomness to prevent the opposite solution from degenerating into a simple mirror image about the center of the dynamic interval, thereby expanding the search coverage while preserving the overall directional

guidance. Because da_d and db_d contract as the swarm converges, the OBL region adapts accordingly, ensuring that opposite solutions are generated within the effective search area.

2.2.4 Adaptive Scout Bee Perturbation

Even with OBL, some wolves may remain stagnant at suboptimal positions for multiple consecutive iterations. Inspired by the scout bee mechanism in the Artificial Bee Colony algorithm (ABC) [11], EDSG-GWO monitors each individual's improvement history and intervenes when no progress is detected. Unlike the original ABC mechanism that completely discards a stagnant solution, this strategy relocates the wolf near the population mean and applies a Gaussian perturbation. The perturbation is formulated as Eq. (12):

$$x_{\text{new}} = \overline{x} + \sigma \cdot (ub - lb) \cdot \text{randn}, \quad \sigma = \sigma_0 \cdot \left(1 - \frac{t}{T}\right) \tag{12}$$

In Eq. (12), $\overline{x}$ is the current population mean, $(ub - lb)$ is the search range in each dimension and serves to scale the perturbation to the problem size, σ_0 sets the base perturbation scale, and randn is drawn from a standard normal distribution. The perturbation amplitude decreases linearly with iterations, providing large perturbations in the early stage and smaller adjustments in the later stage. Two safeguards protect high-quality solutions: (i) the scout mechanism activates only while $\frac{t}{T} < 0.85$, leaving the final 15% of iterations free for undisturbed convergence; (ii) within each activated iteration, only wolves ranked in the bottom 60% by fitness and flagged as stagnant are perturbed. A perturbed position replaces the old one only if its fitness improves.

2.2.5 Algorithm Procedure

Combining the four modules, the complete EDSG-GWO loop proceeds as follows:

Step 1: Initialize N wolves randomly within the search space and set the convergence factor a to 2.

Step 2: Evaluate every wolf's fitness; designate the three best as α, β and δ.

Step 3: Select the top 10% individuals by fitness and generate their opposite solutions via Eq. (11) within the dynamically updated population bounds $[lb, ub]$. Merge the original and opposite populations, retain the best N individuals, and refresh the α, β and δ leadership.

Step 4: Compute the current nonlinear convergence factor, the three leader weights, and the contraction coefficient.

Step 5: For each remaining wolf, compute the weighted blend $\vec{x}^*$ of the three leaders according to Eq. (8), scale the displacement by the contraction coefficient, and clip to the feasible domain.

Step 6: If the iteration ratio $\frac{t}{T} < 0.85$, identify stagnant wolves ranked in the bottom 60% by fitness and apply the Gaussian perturbation directed toward the mean defined in Eq. (12). Accept the perturbed position only if its fitness improves.

Step 7: Re-evaluate all wolves; promote any individual that now surpasses a current leader to the corresponding α, β or δ rank.

Step 8: If $t = T$, output α position as the final solution; Otherwise t++ and return to **Step 2**.

The complete flowchart of the EDSG-GWO algorithm is illustrated in Fig. 1.

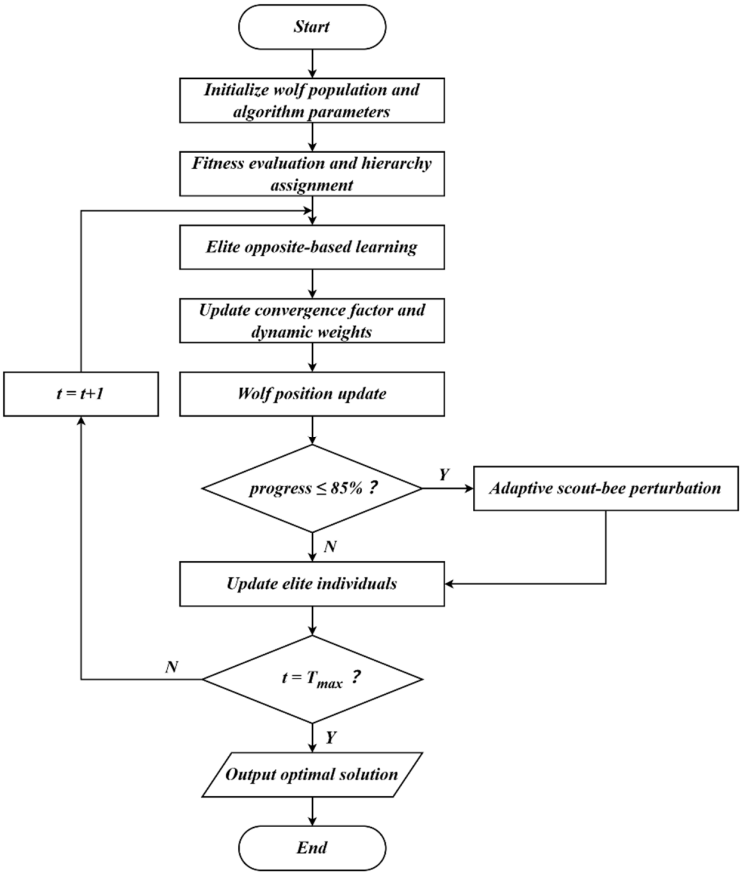


Figure 1: Flowchart of the EDSG-GWO algorithm.

3 Data Preprocessing and Rule Extraction

3.1 PIMA Dataset Preprocessing

The PIMA Indians Diabetes Database comprises 768 records described by eight clinical attributes: Pregnancies (A1), Glucose (A2), Blood Pressure (A3), Skin Thickness (A4), Insulin (A5), BMI (A6), Diabetes Pedigree Function (A7), and Age (A8), with a binary label (1 = diabetic, 0 = non-diabetic). Fuzzy C-Means clustering (m = 2) is applied independently to each attribute, partitioning it into three soft clusters ("Low", "Medium", "High") and producing a 768 × 3 membership matrix per attribute. The iteration terminates when the change between successive membership matrices falls below 1×10^{-7}. Concatenating all eight matrices yields a 768 × 24 fuzzy input matrix. Class labels are one-hot encoded as [1, 0] (diabetic) and [0, 1] (non-diabetic) to match a two-neuron output layer.

3.2 BPNN Workflow

The fuzzy input matrix is randomly split into a 75% training set and a 25% testing set, both L2-normalized to unit length so that no single feature dominates gradient updates. A compact three-layer BPNN is adopted: 24 input neurons mirror the fuzzy feature dimensions, 4 hidden neurons strike a balance between model capacity and computational efficiency, and 2 output neurons encode the class probabilities. Training employs mini-batch gradient descent, with batch size 128, learning rate 0.008, and Sigmoid activation in the hidden layer. An L1 penalty with coefficient $\lambda_{L1} = 0.001$ is imposed on all weights to promote sparsity, a

property that later facilitates clean rule extraction. Each training run spans 100 epochs with 5000 mini-batch gradient updates per epoch, yielding 500,000 parameter updates in total.

The total number of trainable parameters in the network is computed by Eq. (13):

$$N_w = 24 \times 4 + 4 \times 2 = 104 \tag{13}$$

Rather than initializing them randomly, EDSG-GWO searches the 104-dimensional weight space with the training-set total error as its fitness function. The fitness function is defined as Eq. (14):

$$f(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \left(y_{ij} - \hat{y}_{ij} \right)^2 + \lambda_{L1} \sum_{l=1}^{N_w} |w_l| \tag{14}$$

where N is the number of training samples, K is the number of output neurons, y_{ij} and $\hat{y}_{ij}$ are the true and predicted values of the $j-$th output for the $i-$th sample, respectively, and w_l denotes each trainable weight in the network. The optimizer returns a high-quality starting point; standard back propagation then fine-tunes it to convergence.

3.3 WFPRs Extraction

Once the BPNN has been trained, its internal weight matrices encode the discriminative relationships between fuzzy attributes and class labels in a compressed, opaque form. The purpose of this step is to decode those relationships into transparent IF-THEN rules annotated with numerical importance weights. Five operations accomplish the transformation:

(1) Pruning. Any entry in W_1 or W_2 whose absolute value does not exceed 0.5 is zeroed out, yielding sparse matrices $W_{1'}$ and $W_{2'}$. The L1 penalty applied during training has already driven most marginal connections close to zero, so the pruning threshold removes little useful information while greatly simplifying the subsequent rule structure.

(2) Importance matrix. The importance matrix W is computed as transposing the product of the pruned weight matrices, as shown in Eq. (15):

$$W = \left(W_{1'} \cdot W_{2'} \right)^T \tag{15}$$

This yields a 2×24 matrix W, each row corresponding to one output class and each column to one fuzzy attribute. A positive entry signals that the attribute supports the class; a negative entry signals opposition; a zero entry means the attribute was deemed irrelevant by the pruning step.

(3) Rule assembly. For each class, positive entries in W produce "A_n is $A_{n,k}$ $[|w|]$" clauses and negative entries produce "$A_n$is NOT $A_{n,k}$ $[|w|]$" clauses. If a feature retains multiple nonzero subsets, combinatorial expansion generates one rule per subset.

(4) Confidence scoring. For each candidate rule R_k, the confidence CF_k is computed by summing the weighted contribution of every antecedent clause. Let w_j denote the importance weight of the $j-$th clause (taken from the corresponding nonzero entry of W) and u_j denote its membership degree. The contribution c_j of each clause is defined as:

$$c_j = \begin{cases} w_j \cdot \mu_j, & w_j > 0 \\ |w_j| \cdot (1 - \mu_j), & w_j < 0 \end{cases} \tag{16}$$

and the total confidence of rule R_k is computed as

$$CF_k = \sum_{j=1}^{m} c_j \tag{17}$$

where m is the number of antecedent clauses in R_k. For a positive weight, a high membership degree directly supports the rule; for a negative weight, the clause is expressed in the "NOT" form and its low membership degree supports the rule. Rules whose total confidence falls below the threshold $\theta = 0.5$ are discarded; only those with $CF_k \geq 0.5$ participate in the subsequent classification inference.

(5) Classification inference. For a test instance, the total confidence for each class S_l is computed as aggregating the confidences of all activated rules pointing to class C_l:

$$S_l = \sum_{R_k \in \mathcal{R}_l,\, CF_k \geq \theta} CF_k \tag{18}$$

where $\mathcal{R}_l$ is the set of rules whose consequent is C_l. The instance is then assigned to the class with the largest total confidence:

$$\hat{y} = argmax_{l \in \{1,2,\ldots,L\}} S_l \tag{19}$$

4 Experiments and Results

4.1 Benchmark Function Experiments

To verify the general optimization capability of EDSG-GWO before applying it to the BPNN weight optimization task, this subsection evaluates its performance on a set of classical benchmark functions. The comparison algorithms include ABC, standard GWO, DEGWO [12], AGWO [13] and OBLFA-GWO [14], which are the same as those used in the subsequent BPNN optimization experiments. All experiments run on an Intel Core i7-9750H CPU at 2.60 GHz with 8 GB RAM.

4.1.1 Test Functions and Parameter Settings

Twelve benchmark functions are selected for evaluation, as listed in Table 1. The first nine are classical test functions categorized into two groups: unimodal functions (F1–F4) and multimodal functions (F5–F9). Unimodal functions contain a single global optimum with smooth fitness landscapes and are primarily used to assess convergence precision and stability. Multimodal functions possess numerous local optima and complex search spaces, serving as effective benchmarks for evaluating global exploration capability and the ability to escape local optima traps. To further examine algorithmic robustness, three functions (F10–F12) are adopted from the CEC 2017 benchmark suite [15], which feature shifted optima, rotated search spaces, and ill-conditioned landscapes, posing additional challenges beyond classical test functions.

To ensure fairness, the common experimental parameters are set as follows: population size N = 30, maximum iterations T = 2000, and dimension D = 30. Each experiment is independently repeated 30 times, and the mean (Mean) and standard deviation (Std) of the 30 runs are reported as evaluation metrics. The algorithm-specific parameters follow the settings in Table 2.

Table 1: Twelve benchmark test functions.

No.	Function Name	Search Range	Optimum
F1	$f(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]$	0
F2	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]$	0
F3	$f(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]$	0
F4	$f(x) = \sum_{i=1}^{D} i \cdot x_i^4 + \text{rand}[0, 1)$	$[-1.28, 1.28]$	0
F5	$f(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	$[-100, 100]$	0
F6	$f(x) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	$[-10, 10]$	0
F7	$f(x) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	$[-5.12, 5.12]$	0
F8	$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600, 600]$	0
F9	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0
F10	$f(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	$[-100, 100]$	0
F11	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5i \;\; x_i \right)^2 + \left(\sum_{i=1}^{D} 0.5i \;\; x_i \right)^4$	$[-100, 100]$	0
F12	$f(\mathbf{x}) = \sum_{i=1}^{D} \left(10^6\right)^{\frac{i-1}{D-1}} x_i^2$	$[-100, 100]$	0

Table 2: Parameter settings of each algorithm.

Algorithm	Parameters
ABC	$Limit = 50$
DEGWO	$F_{low} = 0.2, F_{high} = 0.8, CR = 0.2$
AGWO	$w = 10, \gamma = 0.95, \varepsilon = 0$
OBLFA-GWO	$epsilon = 0.2, phi_0 = 1.0, gamma = 1.0$
EDSG-GWO	$Limit = 20, \sigma_0 = 0.5$

4.1.2 Experimental Results and Analysis

The statistical results of all six algorithms on the 12 test functions are presented in Table 3, and the corresponding fitness convergence curves are shown in Fig. 2.

As shown in Table 3, EDSG-GWO demonstrates highly competitive performance across all twelve benchmark functions. On the unimodal functions F1–F3, both EDSG-GWO and OBLFA-GWO achieve the global optimum of 0.00E+00 with zero standard deviation. AGWO obtains near-zero but nonzero residual

errors, while ABC produces the poorest results. On F4, EDSG-GWO achieves the lowest mean of 1.65E−05, slightly outperforming OBLFA-GWO, whereas AGWO fails to match even GWO and DEGWO.

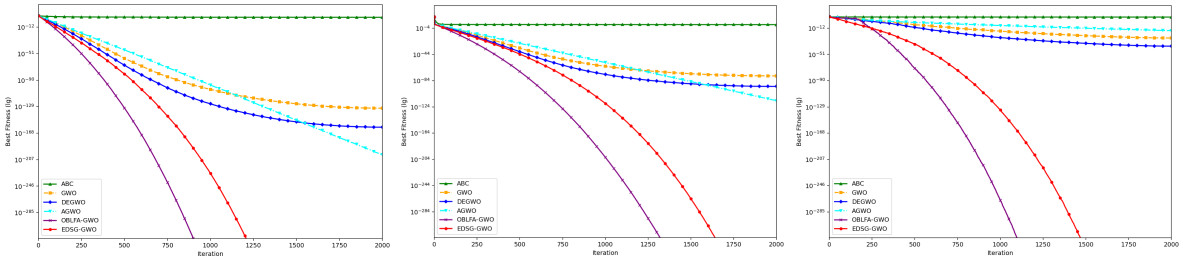
On the multimodal functions F5–F9, EDSG-GWO is the only algorithm to reach 0.00E+00 on both F5 and F6, highlighting its superior local optima escape capability. On F7 and F8, EDSG-GWO and OBLFA-GWO both attain the global optimum, while AGWO yields 1.53E+01 on F7, indicating severe premature convergence. On F9, EDSG-GWO achieves 4.44E−16, marginally outperforming OBLFA-GWO. On the CEC 2017 functions F10–F12, EDSG-GWO and OBLFA-GWO consistently achieve 0.00E+00. AGWO performs well on F10 and F12 but deteriorates sharply on F11, far worse than the standard GWO.

Overall, EDSG-GWO achieves the best or comparable results on all twelve test functions, confirming its effectiveness across unimodal, multimodal, and CEC 2017 landscapes.

The convergence curves in Fig. 2 corroborate these findings. On F1–F4, EDSG-GWO and OBLFA-GWO converge rapidly, whereas GWO, DEGWO, and AGWO plateau at higher fitness values and ABC declines only gradually. On F5 and F6, EDSG-GWO descends sharply and continues refining while OBLFA-GWO stagnates, confirming the benefit of the nonlinear convergence factor and contraction coefficient. On F7–F9, EDSG-GWO and OBLFA-GWO show comparable patterns, both reaching near-optimal values well ahead of the other methods, while AGWO remains trapped at high fitness levels on F7. On F10–F12, EDSG-GWO and OBLFA-GWO again exhibit rapid descent, whereas AGWO converges noticeably slower on F11.

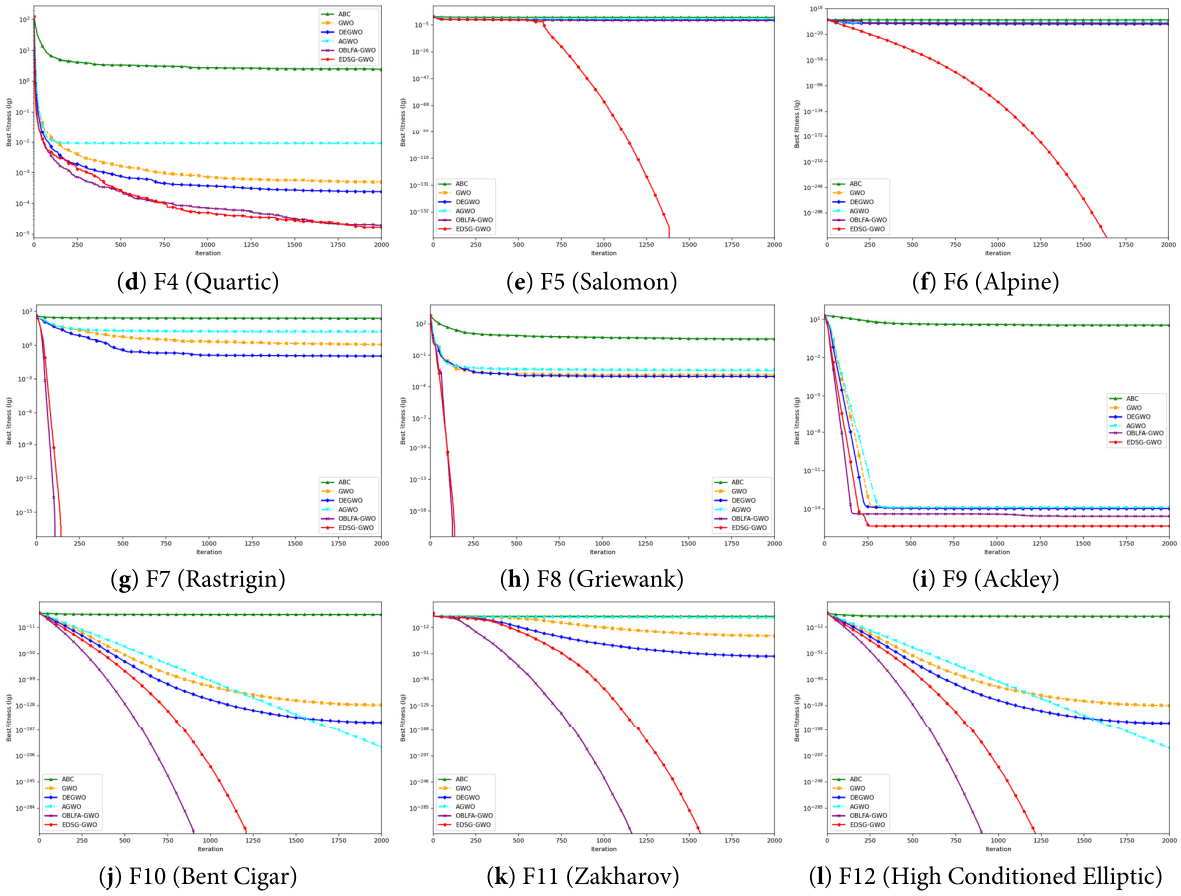
Table 3: Experimental results on 30-dimensional benchmark functions (2000 iterations).

No.	ABC	GWO	DEGWO	AGWO	OBLFA-GWO	EDSG-GWO
	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
F1	4.16e+02 ± 1.65e+02	5.88e−132 ± 3.00e−131	7.05e−160 ± 2.48e−159	1.28e−200 ± 0.00e+00	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00
F2	3.81e+01 ± 1.54e+01	1.94e−77 ± 2.40e−77	1.79e−93 ± 2.94e−93	2.76e−115 ± 4.81e−115	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00
F3	3.55e+04 ± 4.21e+03	2.77e−27 ± 9.49e−27	2.15e−39 ± 1.13e−38	2.16e−16 ± 7.08e−16	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00
F4	2.43e+00 ± 4.72e−01	4.98e−04 ± 3.07e−04	2.40e−04 ± 1.18e−04	8.97e−03 ± 4.20e−03	1.94e−05 ± 1.98e−05	1.65e−05 ± 1.25e−05
F5	5.10e+00 ± 3.29e−01	1.40e−01 ± 4.90e−02	1.23e−01 ± 4.23e−02	2.36e−01 ± 4.71e−02	1.66e−02 ± 3.72e−02	0.00e+00 ± 0.00e+00
F6	2.21e+01 ± 1.59e+00	1.59e−04 ± 3.48e−04	2.65e−05 ± 7.34e−05	1.72e−03 ± 1.82e−03	1.49e−04 ± 5.00e−04	0.00e+00 ± 0.00e+00
F7	2.45e+02 ± 1.42e+01	1.16e+00 ± 3.13e+00	1.08e−01 ± 5.82e−01	1.53e+01 ± 1.25e+01	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00
F8	3.52e+00 ± 1.10e+00	1.37e−03 ± 4.23e−03	9.52e−04 ± 2.94e−03	3.56e−03 ± 6.00e−03	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00
F9	3.65e+00 ± 3.14e−01	1.03e−14 ± 3.27e−15	1.04e−14 ± 3.23e−15	1.25e−14 ± 2.69e−15	2.58e−15 ± 1.74e−15	4.44e−16 ± 0.00e+00
F10	2.37e+08 ± 1.02e+08	1.38e−127 ± 2.72e−127	2.86e−154 ± 9.11e−154	8.34e−194 ± 0.00e+00	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00
F11	6.92e+04 ± 7.02e+03	3.74e−25 ± 1.41e−24	1.11e−55 ± 5.35e−55	2.35e+02 ± 3.41e+02	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00
F12	2.54e+04 ± 1.28e+04	1.34e−129 ± 4.45e−129	1.68e−156 ± 6.95e−156	2.41e−196 ± 0.00e+00	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00



(**a**) F1 (Sphere) (**b**) F2 (Schwefel 2.22) (**c**) F3 (Schwefel 1.2)

Figure 2: (Continued)



(**d**) F4 (Quartic) (**e**) F5 (Salomon) (**f**) F6 (Alpine)

(**g**) F7 (Rastrigin) (**h**) F8 (Griewank) (**i**) F9 (Ackley)

(**j**) F10 (Bent Cigar) (**k**) F11 (Zakharov) (**l**) F12 (High Conditioned Elliptic)

Figure 2: Fitness convergence curves on 30-dimensional benchmark functions. (**a**) F1 (Sphere); (**b**) F2 (Schwefel 2.22); (**c**) F3 (Schwefel 1.2); (**d**) F4 (Quartic); (**e**) F5 (Salomon); (**f**) F6 (Alpine); (**g**) F7 (Rastrigin); (**h**) F8 (Griewank); (**i**) F9 (Ackley); (**j**) F10 (Bent Cigar); (**k**) F11 (Zakharov); (**l**) F12 (High Conditioned Elliptic).

4.1.3 Wilcoxon Rank-Sum Test

To further assess the statistical significance of the performance differences between EDSG-GWO and the comparison algorithms, the two-sided Wilcoxon ranksum test is conducted at a significance level of 0.05. For each test function, the 30 independent runs of EDSG-GWO are compared pairwise against those of every competing algorithm. In Table 4, "+" means EDSG-GWO is significantly better, "−" significantly worse, and "=" no significant difference.

The Wilcoxon rank-sum test results in Table 4 provide rigorous statistical validation of the observed performance differences. EDSG-GWO achieves "+" against ABC and AGWO on all 12 functions, confirming comprehensive superiority across both unimodal and multimodal landscapes. Compared with GWO and DEGWO, EDSG-GWO obtains "+" on 11 and 10 out of 12 functions, respectively, with "=" only on the multimodal Griewank (F8) for both and Rastrigin (F7) for DEGWO, indicating equivalent or superior performance on every test case. Against OBLFA-GWO, the strongest competitor, EDSG-GWO achieves "+" on three functions (F5, F6, F9) and "=" on the remaining nine. Notably, the three significantly better cases are all multimodal functions, demonstrating that the proposed improvements are particularly effective in escaping local optima. On no function does EDSG-GWO receive a "−" against any competitor, confirming that it is never statistically inferior to any compared algorithm.

Table 4: Wilcoxon rank-sum test results on 30-dimensional benchmark functions.

No.	ABC	GWO	DEGWO	AGWO	OBLFA-GWO
F1	+	+	+	+	=
F2	+	+	+	+	=
F3	+	+	+	+	=
F4	+	+	+	+	=
F5	+	+	+	+	+
F6	+	+	+	+	+
F7	+	+	=	+	=
F8	+	=	=	+	=
F9	+	+	+	+	+
F10	+	+	+	+	=
F11	+	+	+	+	=
F12	+	+	+	+	=

4.1.4 Ablation Study

To verify the individual contribution and the synergistic effect of the four proposed improvement strategies, an ablation study is conducted on the same 30-dimensional benchmark functions used in Section 4.1.1. Six configurations are compared: standard GWO as the baseline; four single-strategy variants EDSG-GWO-I to EDSG-GWO-IV, each retaining only one of the strategies—respectively, the nonlinear convergence factor, dynamic weighted position update, elite opposition-based learning, and adaptive scout bee perturbation; and the complete EDSG-GWO integrating all four. Parameter settings follow Section 4.1.1, and each configuration is independently executed 30 times. The mean and standard deviation of the best fitness values are summarized in Table 5.

Table 5: Ablation study results of EDSG-GWO.

No.	GWO	EDSG-GWO-I	EDSG-GWO-II	EDSG-GWO-III	EDSG-GWO-IV	EDSG-GWO
	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std	Mean ± Std
F1	5.88e−132 ± 3.00e−131	4.09e−174 ± 0.00e+00	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00	1.49e−134 ± 7.61e−134	0.00e+00 ± 0.00e+00
F2	1.94e−77 ± 2.40e−77	7.08e−101 ± 2.07e−100	4.43e−264 ± 0.00e+00	0.00e+00 ± 0.00e+00	1.04e−78 ± 1.83e−78	0.00e+00 ± 0.00e+00
F3	2.77e−27 ± 9.49e−27	6.94e−32 ± 2.81e−31	0.00e+00 ± 0.00e+00	1.71e−264 ± 0.00e+00	5.36e−27 ± 1.42e−26	0.00e+00 ± 0.00e+00
F4	4.98e−04 ± 3.07e−04	3.61e−04 ± 2.05e−04	5.74e−05 ± 4.88e−05	1.65e−05 ± 1.42e−05	5.21e−04 ± 2.74e−04	1.54e−05 ± 1.25e−05
F5	1.40e−01 ± 4.90e−02	1.13e−01 ± 3.40e−02	1.11e−01 ± 2.92e−02	0.00e+00 ± 0.00e+00	1.43e−01 ± 4.96e−02	0.00e+00 ± 0.00e+00
F6	1.59e−04 ± 3.48e−04	1.12e−04 ± 4.63e−04	5.90e−06 ± 2.79e−05	0.00e+00 ± 0.00e+00	1.21e−04 ± 3.03e−04	0.00e+00 ± 0.00e+00
F7	1.16e+00 ± 3.13e+00	6.95e−02 ± 3.74e−01	1.58e−01 ± 8.50e−01	0.00e+00 ± 0.00e+00	1.61e+00 ± 3.84e+00	0.00e+00 ± 0.00e+00
F8	1.37e−03 ± 4.23e−03	6.72e−04 ± 3.62e−03	1.03e−03 ± 3.15e−03	0.00e+00 ± 0.00e+00	2.99e−03 ± 5.76e−03	0.00e+00 ± 0.00e+00
F9	1.03e−14 ± 3.27e−15	5.18e−15 ± 1.67e−15	4.00e−15 ± 0.00e+00	4.44e−16 ± 0.00e+00	6.96e−15 ± 1.32e−15	4.44e−16 ± 0.00e+00
F10	1.38e−127 ± 2.72e−127	1.26e−168 ± 0.00e+00	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00	1.04e−129 ± 4.71e−129	0.00e+00 ± 0.00e+00
F11	3.74e−25 ± 1.41e−24	3.93e−39 ± 1.47e−38	0.00e+00 ± 0.00e+00	9.85e−238 ± 0.00e+00	1.70e−29 ± 8.97e−29	0.00e+00 ± 0.00e+00
F12	1.34e−129 ± 4.45e−129	1.22e−170 ± 0.00e+00	0.00e+00 ± 0.00e+00	0.00e+00 ± 0.00e+00	2.90e−132 ± 6.84e−132	0.00e+00 ± 0.00e+00

As shown in Table 5, the four strategies contribute differently to improving standard GWO. EDSG-GWO-III performs most prominently on both unimodal and multimodal functions, achieving theoretical optima on most benchmarks, confirming that elite opposition-based learning is the core mechanism for expanding search coverage and escaping local optima. EDSG-GWO-II ranks second, EDSG-GWO-I demonstrates effective exploration-exploitation balancing, while EDSG-GWO-IV shows limited standalone improvement. When synergistically integrated, the complete EDSG-GWO achieves theoretical optima or

near-optimal precision across all 12 test functions, strictly outperforming every single-strategy variant, demonstrating that the four components compensate for each other's weaknesses across different search phases and function types.

4.1.5 Computational Complexity Analysis

To evaluate the computational efficiency of EDSG-GWO, this section analyzes its time complexity per iteration. Let N denote the population size, D the problem dimension, T_{max} the maximum number of iterations, and ρ the elite ratio ($\rho = 10\%$ in this work). In standard GWO, each iteration involves fitness evaluation $O(N)$ and position update $O(N \cdot D)$, yielding a total complexity of $O(T_{max} \cdot N \cdot D)$.

EDSG-GWO introduces four enhancement strategies upon the standard GWO framework, with the per-iteration overhead analyzed as follows. The nonlinear convergence factor and dynamic weight computation involve only scalar operations and are $O(1)$, thus negligible. The elite opposition-based learning generates and evaluates reverse solutions for $\rho \cdot N$ individuals, incurring a cost of $O(\rho \cdot N \cdot D)$ per iteration; since ρ is typically 0.1, this reduces to $O(N \cdot D)$, equivalent to a single position update in GWO. The adaptive bee-inspired foraging strategy is activated only when the iteration progress ratio falls below $\leq 85\%$ and applies solely to the top 60% of stagnant individuals, resulting in a worst-case cost of $O(N \cdot D)$. Additionally, the merge-and-sort operation for combining the original population with elite reverse solutions requires $O(N \cdot \log N)$, which is dominated by $O(N \cdot D)$.

In summary, the total per-iteration cost of EDSG-GWO is $O(N \cdot D) + O(N \cdot D) + O(N \cdot D) = O(N \cdot D)$, and the overall time complexity remains $O(T_{max} \cdot N \cdot D)$, the same asymptotic order as standard GWO. The constant factor increases by approximately 2 to 3 times, primarily due to the additional fitness evaluations in elite opposition-based learning. Given the significant improvement in convergence accuracy achieved by EDSG-GWO, this modest increase in computational overhead is both reasonable and acceptable.

4.2 BPNN Optimization and Rule Extraction on PIMA Dataset

Having validated the general optimization capability of EDSG-GWO through benchmark function experiments, this subsection further applies it to the practical task of BPNN initial weight optimization and evaluates its effectiveness in improving network training quality and extracting interpretable weighted fuzzy production rules.

4.2.1 Experimental Setup

The comparison algorithms and their parameter configurations follow those specified in Section 4.1.1. For the BPNN weight optimization task, only two settings differ from the benchmark experiments: the population size is reduced to 20 to match the smaller computational budget, and the maximum number of iterations is set to 50 given the higher cost of each evaluation of neural network training. An unoptimized BPNN with random initialization serves as an additional baseline.

4.2.2 Classification Accuracy Analysis

To visually illustrate the convergence behavior of each optimization algorithm during the BPNN initial weight optimization process, Fig. 3 shows the fitness convergence curves of ABC, GWO, DEGWO, AGWO, OBLFA-GWO, and EDSG-GWO over 50 iterations.

As shown in Fig. 3, EDSG-GWO drops rapidly from approximately 0.42 to 0.25 within the first 5 iterations and ultimately converges to approximately 0.19, the lowest value among all methods. AGWO and OBLFA-GWO follow closely, converging to approximately 0.20. GWO and DEGWO converge to

approximately 0.21, while ABC stalls at approximately 0.37. The classification accuracy results on both the training and test sets are summarized in Table 6.

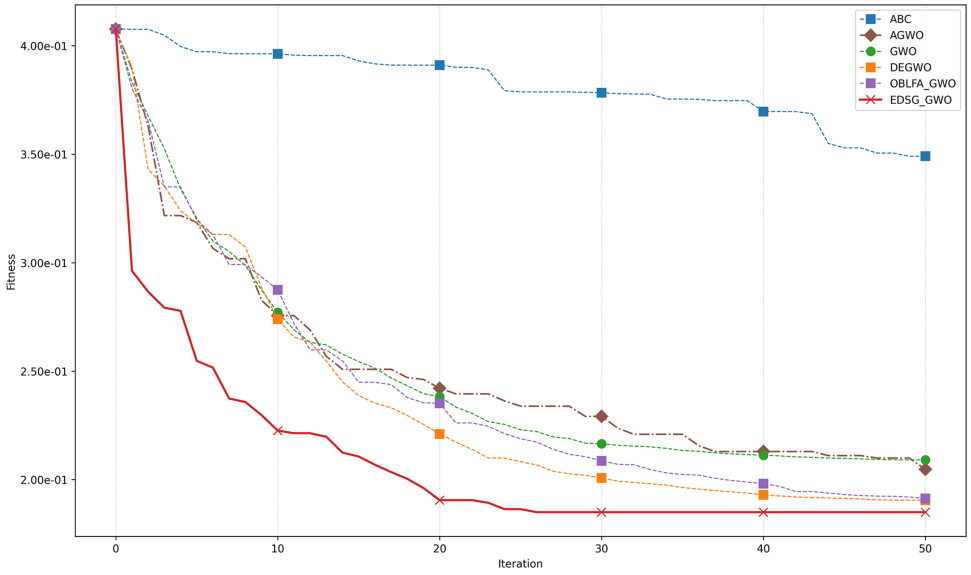


Figure 3: BPNN fitness convergence curves.

Table 6: Classification accuracy of BPNN and WFPRs in PIMA dataset.

Method	Training (BPNN)	Testing (BPNN)	Training (WFPRs)	Testing (WFPRs)
Unoptimized BPNN	77.25%	69.79%	72.57%	70.32%
BPNN optimized by ABC	78.47%	71.35%	74.30%	73.44%
BPNN optimized by GWO	78.65%	73.96%	73.44%	70.83%
BPNN optimized by DEGWO	78.30%	70.31%	73.44%	70.83%
BPNN optimized by AGWO	77.43%	70.31%	74.31%	73.44%
BPNN optimized by OBLFA-GWO	77.78%	73.44%	77.60%	73.44%
BPNN optimized by EDSG-GWO	78.47%	72.92%	79.17%	77.08%

As shown in Table 6, all BPNN models optimized by metaheuristic algorithms outperform the unoptimized baseline on the training set, confirming the effectiveness of initial weight optimization. Among the six methods, GWO achieves the highest BPNN classification accuracy on both training and testing sets, with EDSG-GWO following closely at 78.47% and 72.92%. However, in the more critical WFPRs classification task, EDSG-GWO achieves the best results across all methods, reaching 79.17% on the training set and 77.08% on the testing set, representing improvements of 6.60 and 6.76 percentage points over the unoptimized model. GWO and DEGWO both yield only 70.83% on the WFPRs test set, showing negligible improvement over the baseline, which indicates that higher BPNN accuracy does not necessarily translate into high-quality fuzzy rule extraction. Notably, EDSG-GWO exhibits the smallest training–testing accuracy gap among high-accuracy methods, indicating that its performance gain does not come at the cost of overfitting. These results confirm that the EDSG-GWO–optimized weight matrices possess both good classification ability and structural properties favorable for rule extraction.

4.2.3 Extracted WFPRs

Using the EDSG-GWO optimized BPNN weight matrices, the weight pruning and matrix multiplication procedures described in Eq. (15) are applied sequentially to obtain the importance matrix W, the result of which is presented in Eq. (20).

$$W_{2\times 24} = \begin{bmatrix} 0,0,0,0,-4.1,2.32,0,-1.12,0,0,1.07,0,0,0,0,0,2.03,0,1.26,0,0,1.96,0,0 \\ 0,0,0,0,4.1,-2.32,0,1.11,0,0,-1.06,0,0,0,0,0,-2.03,0,-1.26,0,0,-1.95,0,0 \end{bmatrix} \tag{20}$$

In the importance matrix, each row corresponds to one output class and each column corresponds to one fuzzy input attribute. It is observed that the two rows of the matrix exhibit a highly symmetric structure, with opposite signs for the same attribute positions, reflecting the complementary nature of the two classification categories. Based on the nonzero elements in this matrix, the WFPRs are constructed and presented in Table 7.

Table 7: WFPRs for Class 1 (diabetic) and Class 2 (non-diabetic).

No.	WFPRs
1	IF A2 is NOT A22 [4.1] and A3 is NOT A32 [1.12] and A4 is A42 [1.07] and A6 is A62 [2.03] and A7 is A71 [1.26] and A8 is A81 [1.96] THEN CLASS1
2	IF A2 is A23 [2.32] and A3 is NOT A32 [1.12] and A4 is A42 [1.07] and A6 is A62 [2.03] and A7 is A71 [1.26] and A8 is A81 [1.96] THEN CLASS1
3	IF A2 is A22 [4.1] and A3 is A32 [1.11] and A4 is NOT A42 [1.06] and A6 is NOT A62 [2.03] and A7 is NOT A71 [1.26] and A8 is NOT A81 [1.95] THEN CLASS2
4	IF A2 is NOT A23 [2.32] and A3 is A32 [1.11] and A4 is NOT A42 [1.06] and A6 is NOT A62 [2.03] and A7 is NOT A71 [1.26] and A8 is NOT A81 [1.95] THEN CLASS2

As shown in Table 7, the EDSG-GWO optimized model yields four concise WFPRs, with two rules for each class. In the notation A_{ij}, subscript i denotes the attribute index and j denotes the fuzzy linguistic term ("Low", "Medium", "High") assigned by Fuzzy C-Means clustering. All four rules involve six key clinical features: Glucose (A2), Blood Pressure (A3), Skin Thickness (A4), BMI (A6), Diabetes Pedigree Function (A7), and Age (A8), while Pregnancies (A1) and Insulin (A5) are entirely pruned, indicating negligible contributions after L1 regularized training. The weight distribution exhibits a clear hierarchy: Glucose carries the largest absolute weight of 4.10, followed by BMI, Age, Diabetes Pedigree Function, Blood Pressure, and Skin Thickness. This ranking aligns well with established clinical knowledge, where glucose concentration, BMI, and age are widely recognized as primary risk factors for diabetes. The two classes exhibit a symmetric complementary structure, with positive membership conditions in one class mirrored by negated conditions in the other, enabling clinicians to directly understand how each indicator contributes to the diagnosis.

4.3 Generalization Validation on Breast Cancer Dataset

To evaluate the generalizability of the proposed EDSG-GWO–BPNN–WFPR framework beyond diabetes diagnosis, this subsection applies the identical pipeline to the Breast Cancer Wisconsin dataset. This dataset contains 683 samples with 9 integer-valued cytological features, each ranging from 1 to 10, and a binary class label indicating benign or malignant tumors. Following the same preprocessing procedure described in Section 3, each feature is partitioned into three fuzzy subsets via Fuzzy C-Means clustering,

yielding a 27-dimensional fuzzy input matrix. The BPNN architecture is accordingly configured as 27–4–2, resulting in a 116-dimensional weight search space. All algorithm parameters and experimental settings remain consistent with those specified in Section 4.2. The classification results are reported in Table 8, and the weighted fuzzy production rules extracted from the EDSG-GWO optimized BPNN are listed in Table 9.

Table 8: Classification accuracy of BPNN and WFPRs in Breast Cancer Wisconsin dataset.

Method	Training (BPNN)	Testing (BPNN)	Training (WFPRs)	Testing (WFPRs)
BPNN optimized by ABC	95.70%	94.15%	91.21%	89.47%
BPNN optimized by GWO	96.48%	94.74%	90.42%	87.13%
BPNN optimized by DEGWO	96.48%	94.15%	89.26%	86.55%
BPNN optimized by AGWO	96.68%	97.08%	86.33%	85.96%
BPNN optimized by OBLFA-GWO	96.48%	96.49%	91.21%	88.89%
BPNN optimized by EDSG-GWO	96.88%	97.66%	95.12%	94.15%

Table 9: WFPRs for class 1 (benign) and class 2 (malignant).

No.	WFPRs
1	IF A1 is NOT A12 [1.94] and A2 is A22 [1.0] and A3 is A33 [1.9] and A4 is NOT A42 [1.38] and A5 is A52 [1.27] and A6 is NOT A61 [1.21] and A7 is NOT A72 [1.42] and A8 is NOT A82 [1.14] THEN CLASS1
2	IF A1 is NOT A12 [1.94] and A2 is A22 [1.0] and A3 is A33 [1.9] and A4 is NOT A42 [1.38] and A5 is A52 [1.27] and A6 is A62 [3.88] and A7 is NOT A72 [1.42] and A8 is NOT A82 [1.14] THEN CLASS1
3	IF A1 is A12 [1.94] and A2 is NOT A22 [0.99] and A3 is NOT A33 [1.9] and A4 is A42 [1.38] and A5 is NOT A52 [1.26] and A6 is A61 [1.21] and A7 is A72 [1.42] and A8 is A82 [1.14] THEN CLASS2
4	IF A1 is A12 [1.94] and A2 is NOT A22 [0.99] and A3 is NOT A33 [1.9] and A4 is A42 [1.38] and A5 is NOT A52 [1.26] and A6 is NOT A62 [3.88] and A7 is A72 [1.42] and A8 is A82 [1.14] THEN CLASS2

As shown in Table 8, EDSG-GWO achieves the highest accuracy across all four metrics, with BPNN testing accuracy of 97.66% and WFPRs testing accuracy of 94.15%, outperforming the second-best method by 4.68 percentage points on WFPRs testing accuracy. The training–testing gap of EDSG-GWO on WFPRs is only 0.97 percentage points, the smallest among all methods, indicating strong generalization without overfitting. In contrast, AGWO achieves competitive BPNN testing accuracy but yields the lowest WFPRs accuracy, further confirming that BPNN classification performance alone does not guarantee effective rule extraction. As presented in Table 9, the EDSG-GWO–optimized BPNN produces four compact rules involving only eight of the nine features, demonstrating that the framework can identify a concise and interpretable rule set for breast cancer diagnosis.

5 Conclusions and Future Work

This paper proposed EDSG-GWO, an improved grey wolf optimizer integrating a nonlinear convergence factor, dynamic weighted position update, elite opposite-based learning, and adaptive scout bee perturbation to alleviate BPNN initialization sensitivity and enhance rule interpretability. Experiments on twelve benchmark functions demonstrated competitive or superior optimization performance compared

with five representative algorithms, with no statistically inferior results in the Wilcoxon rank-sum test. On the PIMA dataset, the optimized BPNN achieved a test accuracy of 72.92%, comparable to other metaheuristic-based approaches. More notably, the extracted WFPRs reached a testing accuracy of 77.08%, outperforming all comparison methods and improving the baseline by 6.76 percentage points, with a training–testing gap of only 2.09 percentage points. The four extracted rules involve six clinically relevant features whose weight distributions are consistent with established medical knowledge, effectively transforming the black-box neural network into an interpretable rule-based system. Further validation on the Breast Cancer Wisconsin dataset yielded a WFPRs testing accuracy of 94.15%, confirming the framework's generalizability.

Future work will focus on extending the framework to additional medical datasets, jointly optimizing network topology and weights, and developing adaptive clustering strategies for automatic fuzzy subset determination.

Acknowledgement: Not applicable.

Funding Statement: This research is supported by the National Natural Science Foundation of China (No. 62066016), the Natural Science Foundation of Hunan Province of China (No. 2024JJ7395), the Scientific Research Project of Education Department of Hunan Province of China (No. 24B0481), the Liye Qin Bamboo Slips Research Special Project of Jishou University (No. 25LYY03), the International and Regional Science and Technology Cooperation and Exchange Program of the Hunan Association for Science and Technology (No. 025SKX-KJ-04), and the Postgraduate Scientific Research Innovation Project of Hunan Province (No. CX20251611).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Kai-Qing Zhou; methodology, Xue-Wei Liu and Kai-Qing Zhou; software, Xue-Wei Liu and Shao-Qiang Ye; validation, Xue-Wei Liu and Feng Qin; formal analysis, Xue-Wei Liu; writing—original draft preparation, Xue-Wei Liu, Shao-Qiang Ye and Feng Qin; writing—review and editing, Kai-Qing Zhou. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available. Two datasets are used in this study. The first is the PIMA Indians Diabetes Dataset, originally from the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK), publicly available on Kaggle at https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database. The second is the Breast Cancer Wisconsin (Original) Dataset, originally from the University of Wisconsin Hospitals, publicly available on the UCI Machine Learning Repository at https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ye S, Zhou K, Zain AM, Wang F, Yusoff Y. A modified harmony search algorithm and its applications in weighted fuzzy production rule extraction. Front Inf Technol Electron Eng. 2023;24(11):1574–90. doi:10.1631/FITEE.2200334.
2. Nilashi M, Ibrahim O, Ahmadi H, Shahmoradi L. A knowledge-based system for breast cancer classification using fuzzy logic method. Telemat Inform. 2017;34(4):133–44. doi:10.1016/j.tele.2017.01.007.
3. Andrews R, Diederich J, Tickle AB. Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowl Based Syst. 1995;8(6):373–89. doi:10.1016/0950-7051(96)81920-4.
4. Chen WL, Zhou KQ, Sarkheyli-Hägele A, Qin F, Hasikin K, Kang DW, et al. Cross-lingual transfer learning for knowledge graph acquisition: paradigms, resources and challenges. Expert Syst Appl. 2026;303(6):130434. doi:10.1016/j.eswa.2025.130434.

5. Mousavi SM, Abdullah S, Niaki STA, Banihashemi S. An intelligent hybrid classification algorithm integrating fuzzy rule-based extraction and harmony search optimization: medical diagnosis applications. Knowl Based Syst. 2021;220(3):106943. doi:10.1016/j.knosys.2021.106943.
6. Chakraborty M, Biswas SK, Purkayastha B. Rule extraction from neural network trained using deep belief network and back propagation. Knowl Inf Syst. 2020;62(9):3753–81. doi:10.1007/s10115-020-01473-0.
7. Ojha VK, Abraham A, Snášel V. Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng Appl Artif Intell. 2017;60:97–116. doi:10.1016/j.engappai.2017.01.013.
8. Faris H, Aljarah I, Al-Betar MA, Mirjalili S. Grey wolf optimizer: a review of recent variants and applications. Neural Comput Appl. 2018;30(2):413–35. doi:10.1007/s00521-017-3272-5.
9. Li S, Mo LP, Yin B, Min W. An improved sparrow search algorithm and its application in the deployment of 3D wireless sensor nodes. Sci Rep. 2026;16(1):2329. doi:10.1038/s41598-025-32080-0.
10. Zhang Y, Cai Y. Adaptive dynamic self-learning grey wolf optimization algorithm for solving global optimization problems and engineering problems. Math Biosci Eng. 2024;21(3):3910–43. doi:10.3934/mbe.2024174.
11. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim. 2007;39(3):459–71. doi:10.1007/s10898-007-9149-x.
12. Wang H, Zhu J, Li W. An improved back propagation neural network based on differential evolution and grey wolf optimizer and its application in the height prediction of water-conducting fracture zone. Appl Sci. 2024;14(11):4509. doi:10.3390/app14114509.
13. Meidani K, Hemmasian A, Mirjalili S, Barati Farimani A. Adaptive grey wolf optimizer. Neural Comput Appl. 2022;34(10):7711–31. doi:10.1007/s00521-021-06885-9.
14. Tang H, Xu Y, Lin A, Heidari AA, Wang M, Chen H, et al. Predicting green consumption behaviors of students using efficient firefly grey wolf-assisted K-nearest neighbor classifiers. IEEE Access. 2020;8:35546–62. doi:10.1109/ACCESS.2020.2973763.
15. Wu GH, Mallipeddi R, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization [Internet]. [cited 2026 Jan 1]. Available from: https://www.researchgate.net/publication/317228117.