

PERFORMANCE ANALYSIS OF EBGWO ALGORITHM BASED ON BENCHMARK FUNCTION AND CREDIT RISK DATASET

SUIHAI CHEN^{1*}, BONG CHIH HOW², PO CHAN CHIU³

Abstract: *This study analyzes the EBGWO algorithm, and uses 20 benchmark test functions, which are divided into single-peak and multi-peak functions to test algorithm performance and cites WOA algorithm for in-depth analysis and comparison. The research involves benchmark function, discusses its development and exploration stage, and applies EBGWO algorithm to real credit risk data set for testing. The outcomes demonstrates that the EBGWO algorithm performs more comprehensively than GWO algorithm method.*

INTRODUCTION

An adaptive GWO algorithm (EB-GWO) is proposed in this paper, it is combined with CatBoost algorithm to build a new algorithm. The method uses adaptive leadership hierarchy (EB) to further optimize the Grey Wolf algorithm (GWO), which is then combined to enhance the convergence GWO's performance. The inefficiency of the GWO is avoided. Achieving accuracy enhancement involves striking a balance between the capabilities of local search and global development. GWO and CatBoost algorithms are chosen to improve model performance because GWO possesses simple structure, convenient implementation, and few adjustment parameters. CatBoost algorithm exhibits good accuracy, interpretability, and stability.

GWO possesses a merit in its straightforward structure, accompanied by convenient implementation and a minimal number of adjustment parameters. GWO also has its advantages: the algorithm model is novel; the optimization ability is more prominent when the algorithm does not fall into premature maturity. Therefore, GWO has attracted the attention of many scholars with its good performance since it was proposed. Nevertheless, because practical optimization issues are so complex, various improved GWO algorithms have been created to enhance the optimization capabilities of the original version, with a focus on achieving faster global convergence speeds, Mehak introduced “chaos theory” into GWO, which is called CGWO [2]. To attain the objective of achieving a balanced exploration and

¹ Dr, Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS), Sarawak, Borneo, Malaysia, 943000.

ORCID iD: <https://orcid.org/0000-0003-4888-8726>, Email: c51772109@163.com

² Professor, Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS), Sarawak, Borneo, Malaysia, 943000

ORCID iD: <https://orcid.org/0000-0002-2612-3493>, Email: chbong@unimas.my

³ Professor, Computer Science and Information Technology, Universiti Malaysia Sarawak (UNIMAS), Sarawak, Borneo, Malaysia, 943000

ORCID iD: <https://orcid.org/0000-0003-0414-7926>, Email: pcchiu@unimas.my

*Correspondence: c51772109@163.com

development capabilities, Joshi proposed an enhanced GWO algorithm (EGWO) with a well-optimized mechanism [12]. Ibrahim suggested an enhanced GWO algorithm with the intention of harmonizing the capabilities of the original version [11]. Due to the GWO algorithm (COGWO2D) limitations, such as poor precision and a sluggish convergence speed, GWO combined with the PSO algorithm (PSO-GWO) was proposed for the weighted distance optimization problem [5]. The grey wolf weighted distance optimization algorithm (WD-GWO) was proposed, it updates a single position by using the weighted total of the best places rather than a simple average [10]. Simulation experiments verify the superiority of the algorithm. Powell's local optimization technique served as the foundation for PGWO, an expanded GWO algorithm that was presented [15]. Powell's local optimal method was a method of calculating the minimum value in the absence of function differentiability. Additionally, the experimental results demonstrated its superiority. In terms of application, the algorithm finds application in domains like machine learning. For example, Kamboj proposed sequential hybridization of GWO and PSO to optimize single-region unit combination problems [6]. However, because of the lengthy sequence execution time, the approach converges gradually. GWO and a local search algorithm have been integrated by Mahdad and Srairi to handle protected smart grid power system management in emergency scenarios [7]. Compared with other optimization methods, the results show a promising application of this method.

Therefore, on the basis of discussing the number of leaders missing in the GWO, this paper proposes an adaptive GWO (EB-GWO for short) and balances the exploration and development capabilities of GWO. This model is used to improve the original GWO. The Big Wolf algorithm's inefficiencies are somewhat avoided, and the algorithm's capacity for both local search and global expansion is balanced to increase algorithmic accuracy.

ADAPTIVE LEADERSHIP GRAY WOLF OPTIMIZATION ALGORITHM (EB-GWO)

EB-GWO Algorithm

The GWO algorithm gray wolves rely primarily on α , β , and δ information to find prey, the number of leading wolves (α , β , δ) is fixed to 3, which leads to weak algorithm's exploration ability and easy to fall into local optimal solution [3]. In order to find a solution for this problem, this study suggests a method to adaptively update the number of leader wolves. The algorithm determines the number of leader wolves according to the actual situation to stop the falling of algorithm into the optimal solution, and the optimized GWO algorithm is called EBGWO algorithm. The main contribution of EBGWO is to use different numbers of leaders to update a single position in the iteration process, and to continuously adjust the number of leader wolves through the parameterized adaptive mechanism to achieve an exploration and exploitation capabilities balance.

Quantity Renewal Based on Leadership

α , β and δ are the three leaders' wolves that the original GWO algorithm is count of. The original wolf's position is represented by $X(t+1)$ in this procedure which is determined by averaging X_1 , X_2 and X_3 according to mathematical formula. But the algorithm is gain by changing by changing the number of leading wolves when dealing

with the big sets that have irregular samples, when differentiating between good and bad clients, or when distinctions are not quite evident. For example, tow lead wolves (X1 and X2) are sufficient if the data facilitates simple consumer categorization, increasing convergence speed and lowering iterations. However, when the data lies in the middle of distinct categories, calling for a more expansive search radius, the number of leaders must be increased, resulting in the creation of fourth leader X4. GWO's original algorithm application of a three leader wolves under all data situation results in inefficiency, necessitating additional optimization.

In order to address this issue, a new update formula is introduced by making the x leaders number four; α guides β , δ and γ , and all these represents four optimal solutions. The new updated formula is as follows;

$$\begin{aligned} D_{\alpha} &= |C_{1\alpha}X_{\alpha} - X, \\ D_{\beta} &= |C_{2\alpha}X_{\beta} - X, \end{aligned} \quad \text{Equation 2A}$$

$$\begin{aligned} D_{\gamma} &= |C_{4\alpha}X_{\gamma} - X, \\ X_1 &= X_{\alpha} - A_{1\alpha}D_{\alpha}, \\ X_2 &= X_{\beta} - A_{2\alpha}D_{\beta}, \end{aligned} \quad \text{Equation 2B}$$

$$\begin{aligned} X_3 &= X_{\delta} - A_{3\alpha}D_{\delta}, \\ X_4 &= X_{\gamma} - A_{4\alpha}D_{\gamma}. \\ X(t+1) &= \frac{X_1 + X_2 + X_3 + X_4}{4} \end{aligned} \quad \text{Equation 2C}$$

The distances between wolves ($D_{\alpha}, D_{\beta}, D_{\delta}, D_{\gamma}$), their present locations ($X_{\alpha}, X_{\beta}, X, X_{\gamma}$), Synergy coefficients (A and C) and random vectors (r1 and r2) are included in this calculation.

$$\begin{aligned} D_{\alpha} &= |C_{1\alpha}X_{\alpha} - X, \\ D_{\beta} &= |C_{2\alpha}X_{\beta} - X, \end{aligned} \quad \text{Equation 2D}$$

$$\begin{aligned} X_1 &= X_{\alpha} - A_{1\alpha}D_{\alpha}, \\ X_2 &= X_{\beta} - A_{2\alpha}D_{\beta}, \end{aligned} \quad \text{Equation 2E}$$

$$X(t+1) = \frac{X_1 + X_2}{2} \quad \text{Equation 2F}$$

In a similar vein letting go of a leader means letting α guide β , and modifying the update formula correspondingly:

$X(t=1)$ is dynamically calculated based on actual conditions in scenarios where the number of leaders' wolves fluctuates between four and 2. Depending on the value of $|A|$, dynamic modifications can be made by substituting alternatives for the fixed formula used in original GWO algorithm such as substituting $X(t+1) = (X_1 + X_2)/2$ with $X(t+1) = (X_1 + X_2 + X_3)/3$ OR $X(t+1) = (X_1 + X_2 + X_3 + X_4)/4$. Increasing the leader wolves is required for comprehensive investigation when $|A| \geq 1$, indicating a bad solution or ambiguous data, it means that the solution is not optimal and the three classes of GWO algorithm needs calculation. the calculations show the blurry differentiation between good and bad customers and the more calculation required so that the wolves number increased. On the other hand, decreasing leader wolves improves algorithm efficiency when $|A| < 1$ and the bad solution is comprehensively superior. We obtain the best convergence ability [4], only by fluctuating the number of wolves to minimum and maximum, which will affect the original GWO algorithms' convergence. Hence, this research is based on decreasing and increasing the number of wolves through original GWO algorithm. Leader wolves will increase in number of 1, $X(t+1)$ is four leader wolves in case of $A > 1$, and the leader wolf decrease in number of 1, $X(t+1)$ is two leader wolves in case of $A < 1$. The updated formula of GWO algorithm is shown in below (2G)

$$X(t+1) = \begin{cases} \frac{X_1 + X_2 + X_3 + X_4}{2} & \text{if } |A| > 1 \\ \frac{X_1 + X_2}{2} & \text{else} \end{cases} \quad \text{Equation 2.7}$$

Pseudocode of EBGWO

The EBGWO algorithm's pseudocode is shown in Figure 1. The new EBGWO algorithm suggested in this study intends to enhance the exploration and exploitation balance, in contrast to the GWO method, which fixes the number of leader wolves (α , β , δ) at three, resulting in weak exploration capabilities. This is accomplished by using different number of leaders to update a single position during the course of iterations. The algorithm's step 10 to 14 involved conditional decisions and the execution of matching arithmetic expressions. The number of leader wolves is continuously changed by a parameterized adaptive process in an effort to improve the algorithm's poor exploration capability. This method aims to enhance the algorithm's overall performance in practical situations while maximizing its exploration capabilities.

Algorithm	EBGWO
1:	Initialize the grey wolf population $X_i (i=1, 2, \dots, n)$
2:	Initialize a , A , and C
3:	Calculate the fitness of each search agent
4:	X_α = the best search agent
5:	X_β = the best search agent
6:	X_δ = the best search agent
7:	X_γ = the best search agent
8:	while ($t < \text{Max number of iterations}$)
9:	for each search agent
10:	▷ Update the position of the current search agent
11:	if ($ A < 1$):
12:	$X(t+1) = (X_\alpha + X_\beta)/2$
13:	else:
14:	$X(t+1) = (X_\alpha + X_\beta + X_\delta + X_\gamma)/4$
15:	end for
16:	Update a , A , and C
17:	Calculate the fitness of all search agents
18:	Update $X_\alpha, X_\beta, X_\delta$, and X_γ
19:	$t = t + 1$
20:	end while
21:	return X_α

Figure 1: EBGWO algorithm pseudocode.

DATA SET INTRODUCTION

The dataset used in this paper are from real credit databases. The credit loan customer loan information of a domestic financial company from 2017 to 2021, it call "credit". Web is <http://archive.ics.uci.edu/>.

The data set for "a domestic financial company", variable name, type and value distribution are as follow Table 1.

Table 1. Data content Situation

Variable	Value Distribution
Account Balance	1:X<0, 2:0<=X<200, 3:X>200, 4:no checking account
Duration of Credit (month)	Duration in month
Payment Status of Previous Credit	Its value is 0-4. 0: No loans; 1: The loan has been repaid; 2. Timely repayment; 3. Delay in repayment; 4. Other bank loans.
Purpose	Its value is 0-10. 0: New car, 1: old car, 2: Furniture/equipment, 3: Radio/television, 4: Domestic appliances, 5: Repairs, 6: Education, 7: Vacation (nonexistent), 8: Retraining, 9: Business, 10: Other purposes.
Credit Amount	Credit amount
Value Savings/Stocks	1: X< 100, 2:100 <=X< 500, 3:500 <=X< 1000, 4:X>= 1000, 5: unknown/no savings account
Length of current employment	1: unemployed, 2:X< 1, 3:1 <= X< 4, 4:4<=X< 7: 5:>= 7
Instalment per cent	Installment rate in percentage of disposable income
Sex & Marital Status	A91: male: divorced/separated, 2: female: divorced/separated/married, 3 males: single, 4: male: married/widowed, 5: female: single
Guarantors	1: none, 2: co-applicant, 3: guarantor
Duration in Current address	Present residence since
Most valuable available asset	1: Real estate, 2: If not "1" - building life insurance/ society savings agreement, 3: If not "1"/"12" - car or other, not in attribute, 4: No property/ unknown.
Age (years)	Age in years
Concurrent Credits	1: bank, 2: stores, 3: none
Type of apartment	1: rent, 2: own, 3: for free
No of Credits at this Bank	Number of existing credits at this bank
Occupation	1: unemployed/ unskilled - non-resident, 2: unskilled – resident, 3: skilled employee / official, 4: management/highly qualified employee/ officer/ self-employed
No of dependents	Number of people being liable to provide maintenance for
Telephone	1: none, 2: yes, registered under the customers' name
Foreign Worker	1: yes, 1: no

The data set for "Credit_China" is real data, which tends to have missing data for various reasons. In the experiment, the data are processed according to the degree of missing data, items with a large missing rate are deleted, and missing items are filled with different methods such as mean interpolation, feature median filling, and pattern filling that are less than 20%. Then perform "unique hot encoding" and "feature derivation" on the data. Finally, feature selection is performed. Through feature engineering, following data set containing 19 features is obtained.

The German personal credit database is very organized and does not require any trimming. The database contains a total of customer data. All customers are classified as either "good" or "bad." Other characteristic variables that can be used for calculation are: X1(existing checking account), X2(month's duration), X3(Credit history), X4(Purpose), X5(Credit amount), X6 (Savings account/bonds), X7(Present employment since), X8(installation rate as a percentage of disposable income), X9(personal status and sex), X10(other debtors/guarantors), X11(present residence since), X12(property), X13(age in years), X14(other installment plans), X15(housing), X16(number of credits available at this bank), X17(job), X18(number of people responsible for maintaining), X19(telephone) and X20(foreign worker).

Among them, X2(Duration in month), X5(Credit amount), X13(Age in years) is a continuous variable, X8(Installment rate as a disposable income's percentage), X11(Present residence since), X16(existing credits number at this bank) and X18(People responsible for providing maintenance) are discrete variables, and the rest are character variables. Data sets were also tested with improved models.

EXPERIMENTATION AND EVALUATION

Evaluation Index

Loan data are characterized by data imbalances. That is, the sample size of default risk is much smaller than the sample size of normal loans. Therefore, when the classification accuracy is used as the final evaluation index, even if the default sample size is too small, the accuracy will remain high. In this way, the function of predicting loan default customers is lost. However, personal loan defaults often impose high costs on the day-to-day lives of financial firms. Therefore, it is unreasonable to use "accuracy" as the only evaluation index for predicting loan risk in model evaluation. In this paper, the AUC value is utilized as the evaluation index. The normal loan sample is designated as the positive sample, while the default sample is designated as the negative sample. The definition of the confusion matrix is illustrated in Figure 2.

Confusion Matrix			
Actual \ Predict	0	1	
0	TN (true negative)	FN (false negative)	$Precision = \frac{TP}{TP + FP}$
1	FP (false positive)	TP (true positive)	$Recall = \frac{TP}{TP + FN}$
			$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$
			$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$

Figure 2: Confusion Matrix.

Based on the confusion matrix, accuracy (A), precision (P) and recall (R) were defined as follows:

$$A = (TN + TP) / (FN + FP + TN + TP)$$

$$P = TP / (FP + TP)$$

$$R = TP / (FN + TP)$$

According to the above definition, the ROC curve can be obtained to consider the prediction accuracy of unbalanced samples.

Experiment

Two experiments are carried out in this paper: 1. This study adopts 20 benchmark test functions, which are divided into unimodal and functions of multi-modal. The parameters of the algorithm are standardized for a comparison. 2. Apply GWO and EBGWO algorithms to the "China_Credit" dataset to evaluate their algorithm training efficiency. EBGWO demonstrated a shorter run time, demonstrating its adaptability to adjust the lead wolves numbers to improve the algorithm's efficiency.

Table 2. Expressions of Unimodal Functions (F1-F7).

Function Name	Expression	Dimensionality	Range
F1	$F_1(x) = \sum_{i=1}^D x$	30	[-100,100]
F2	$F_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	30	[-10,10]
F3	$F_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^D x \right)$	30	[-100,100]
F4	$F_4(x) = \max_{1 \leq i \leq D} x_i $	30	[-100,100]
F5	$F_5(x) = \sum_{i=1}^D (100(x_{i+1} - x_i^2) - (x_i - 1)^2)$	30	[-30,30]
F6	$F_6(x) = \sum_{i=1}^D (x_i + 0.5)^2$	30	[-100,100]
F7	$F_7(x) = \sum_{i=1}^D ix_i^4 + random(0,1)$	30	[-1.25,1.25]

Table 3. Expressions of Unimodal Functions (F8-F20).

Function Name	Expression	Dimensionality	Range
F8	$F_8(x) = \sum_{i=1}^D x \sin(\sqrt{ x_i })$	30	[-100,100]
F9	$F_9(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i + 10))$	30	[-5.12,5.12]
F10	$F_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D x_i^2 \cos(2\pi x_i) \right) + 20 + e$	30	[-32,32]
F11	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	[-600,600]
F12	$F_{12}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (100(x_i - 1)[1 + \sin^2(3\pi x_{i+1} + 1)] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \} + \sum_{i=1}^{D-1} u(x_i, 5, 100, 4)$	30	[-50,50]
F13	$F_{13}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^2})^{-1}$	2	[-65,65]
F14	$F_{14}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]
F15	$F_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]
F16	$F_{16}(x) = \left(x^2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[-5,5]
F17	$F_{17}(x) = [1 + (x_1 + x_{12} + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2 \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)])]$	2	[-2,2]
F18	$F_{18}(x) = \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 a_{ij} (x_i - p_{ij})^2 \right)$	3	[1,3]
F19	$F_{19}(x) = \sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]
F20	$F_{20}(x) = \sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]

Experimental Result of Benchmarking Test Functions and Analysis: This study adopts 20 benchmark test functions for simulation experiments. Table 2 and Table 3 give the basic information of 20 benchmark functions, and the values in the table are referring to Wu, et al. [13], where F1-F7 is an unimodal function and F8-F20 is a multi-modal function. The algorithm parameters show: N=30 population size, T=1,000 maximum iteration numbers, and the iterations maximum numbers is taken as the termination condition. To ensure the fairness of the comparison, each algorithm runs 100 iterations independently to record the Best value (Best) and standard deviation (Std) and compares with the operation results of GWO and EBGWO. All experiments are implemented in MATLAB-R2012a.

WOA (whale optimization algorithm) and GWO algorithm are both intelligent group algorithms. In intelligent group algorithm testing, WOA algorithm has a better comparison effect [9]. In this study, WOA algorithm is compared with GWO and EBGWO algorithm. The comparison process is as follows.

Attacking prey (development stage): This section compares GWO, EB-GWO and WOA algorithms. Table number 4 shows the Best and Std, the three algorithms' values (GWO, EB-GWO, WOA) in the seven test functions. The table 4 results show that for the best value, EBGWO algorithm this column is better than the original GWO results on 5 test functions (F1-F4 and F6), and most of the better results than WOA results in all functions. This means that the proposed algorithm is closer to the theoretical optimal solution in these test functions, and the development ability is stronger than that of the comparison algorithm, indicating that the development ability of the EB-GWO algorithm is better than that of GWO. In terms of Std values, EB-GWO gets better results on F1, F2, F3, F4, and F7 functions.

Research for prey (Exploration stage): Functions F8-F20 are multimodal functions. They include many local optimizations, and the local number optimizations increases rapidly with the increase in dimension. Therefore, these functions are considered to be very useful in evaluating the exploration capabilities of algorithms [8]. Table 5 lists the Best and Std values of the three comparison algorithms on F8-F20. As the table shows, the Best Value of EB-GWO is the smallest on 10 functions (F9, F11, and F13-F20, Bold font in the table), the Best Value of GWO is the smallest on 8 functions (F9, F11, F13-F18), and the Best Value of WOA is also the smallest on 10 functions (F8-F13, F15-F18). For Std values, EB-GWO has the smallest Std value on 7 functions (F8-F10, F13, F17-F18, F20), GWO has the smallest Std value on 3 functions (F11, F14, F19), and WOA has the smallest Std value on 4 functions (F9, F12, F15-F16). From the analysis of test results, EBGWO is the best in the test results of most test functions, and algorithm that is proposed has strong competitiveness in solving multi-peak problems. This also proves that EB-GWO has good exploration capability.

Table 4. Test Results on Unimodal Functions (F1-F7).

Function Name	GWO	EB-GWO	WOA	GWO	EB-GWO	WOA
	Best	Best	Best	std	std	Std
F1	1.14e-60	-8.70e-74	0	1.32e-59	-1.05e-71	0
F2	2.14e-35	-1.19e-43	0	5.93e-35	-9.00e-43	0
F3	1.42e-19	1.44e-23	495.309	9.91e-15	1.65e-15	7341.480
F4	9.22e-16	7.27e-19	0.00624	1.20e-15	2.94e-17	16.0171
F5	26.1808	26.1901	26.6542	0.7031	0.633317	0.36512
F6	0.24972	8.20e-05	0.00810	0.2950	0.417425	0.07957
F7	0.00018	0.00024	7.00e-05	0.0005	0.000394	0.002148

Table 5. Test Results on Multimodal Functions (F8-F20).

Function Name	GWO	EB-GWO	WOA	GWO	EB-GWO	WOA
	Best	Best	Best	std	std	std
F8	-6722.73	-6732.77	-7708.60	1143.76	631.87	722.90
F9	0	0	0	1.884520	0	0
F10	1.50e-14	7.99e-15	8.88e-16	2.39e-15	0	2.80e-15
F11	0	0	0	0.004278	0.008869	0.016188
F12	0.09489	0.21346	0.08240	0.243788	0.148689	0.080860
F13	0.99801	0.998001	0.99801	0.912259	0.836571	0.958412
F14	0.00031	0.00031	0.00032	1.13e-07	0.000288	0.000491
F15	-1.03163	-10.3163	-1.03163	3.83e-09	4.29e-09	1.36e-10
F16	0.397891	0.397891	0.397891	5.83e-07	2.00e-06	2.48e-07
F17	3	3	3	1.00e-05	4.00e-06	5.00e-06
F18	-3.86278	-3.86278	-3.86278	5.30e-05	6.00e-06	0.000497
F19	-10.4029	-10.40286	-10.40257	0.000225	0.00030	2.811192
F20	-1053604	-10.53636	-10.53573	0.000411	0.000333	3.236052

According to the research of experts [1], the two benchmark functions of sphere and Schaefer can effectively test the performance of local search and global search of intelligent optimization algorithms. The following two benchmark functions will be selected to visually show the GWO and EBGWO development phase and exploration phase.

Two function parameters settings are shown in table 6, which includes values for vector X, population size N, upper and lower boundaries of D and i. Experts Yao, Liu, and Lin's research yielded three parameters, which are essential for the algorithmic performance [14]. Since the EBGWO algorithm is usually configured to have three leaders by default, the majority of experimental changes revolve around varying the number of leaders. With the use of this technique, algorithm behavior under various leadership configuration maybe thoroughly explored, leading to a better understanding of and optimization if algorithmic performance.

Table 6. Sphere and Schaefer Functions Parameter Settings.

Name of Function	Expression	Range	D	N	T _{max}
Sphere	$F_1(x) = \sum_{i=1}^D x_i^2$	[-100,100]	30	20	100
Schwefel	$F_8(x) = \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$	[-100,100]	30	20	100

Experimental Results and Analysis of Shpere and Schwefel Functions:

Initially, because the iterations numbers are unknown, testing are carried out in increments of 0 to 100. These outcomes are then used to inform the selection of iteration counts. Test results using the Shpere and Schwefel functions are in Figure number 3.

Figure 3 makes it clear that the value of A stabilizes as the number of repetitions gets closer to roughly 100. Based on this observation, it seems fair to use 100 iterations for this study. When taking into account the mathematical formula of the GWO algorithm, A has a direct impact on the update position of the leader wolf. As a result, A's stability suggests that the algorithm's leadership is stable.

Solutions obtained from both the GWO and EBGWO algorithms steadily converge towards the ideal solution during their evolutionary process, eventually converging to the optimum global, as can be seen by examining the search histories shown in Figure 3 for both functions.

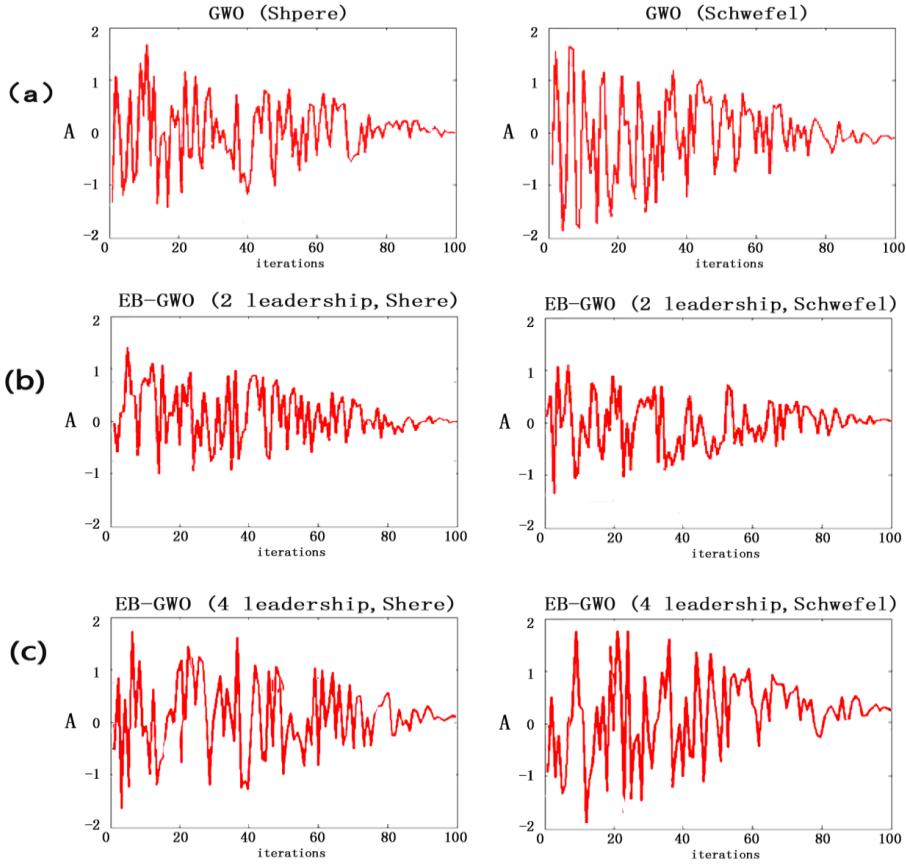


Figure 3: Tests of A in Shpere and Schwefel functions.

A is important since it directly affects the leader wolf' position update. As a result, the wolves' range of movement and update will expand with A's range, which will impact the search's entire scope. When the results that are broken down as highlighted in Figure 3, Row (a) shows the GWO algorithm results with three leader wolves, while Rows (b) and (c) show the results of the EBGWO algorithm with two and four leader wolves, respectively. EBGWO with four leader wolves has a more varied and expansive track distribution across 100 iterations on the Schwefel and Sphere functions, according to a comparison of Rows (b) and (c). This suggests that there is a greater capacity for exploration with more leaders. When the four steps of the GWO algorithm are taken into account, it is noted that the fluctuation amplitude of A in Row (b) is smaller than in Row (c), where $|A| < 1$ indicates wolves attacking prey and displaying strong development capability. Additionally, the majority of the variations in Row (b) are less than 1, indicating a more robust developmental stage in the case with two leader wolves. Row (a) (GWO) is situated between Rows (b) and (c) (EBGWO), suggesting that when using varied leader numbers, the GWO algorithm performs worse than EBGWO in terms of development and exploration. As compared to the experimental most effective results of the algorithms of the GWO (leaders wolves numbers is 3) and the EBGWO (leaders wolves numbers is 2 or 4), row (a) in the Figure 6.3 shows the optimum value estimated as the total

ranges of iterations from zero to twenty and twenty to sixty is a line that is straight, making it simple to lie into the optimal value that is local. As the repetitions numbers exceeds 70, the global best value emerges gradually. Compared to Figure 4 rows (b) and (c) of the EBGWO algorithm, the best value shows that with the total leaders numbers among wolves is two, the best global solution converges in 20 iterations, and the rate of convergence is quite rapid.

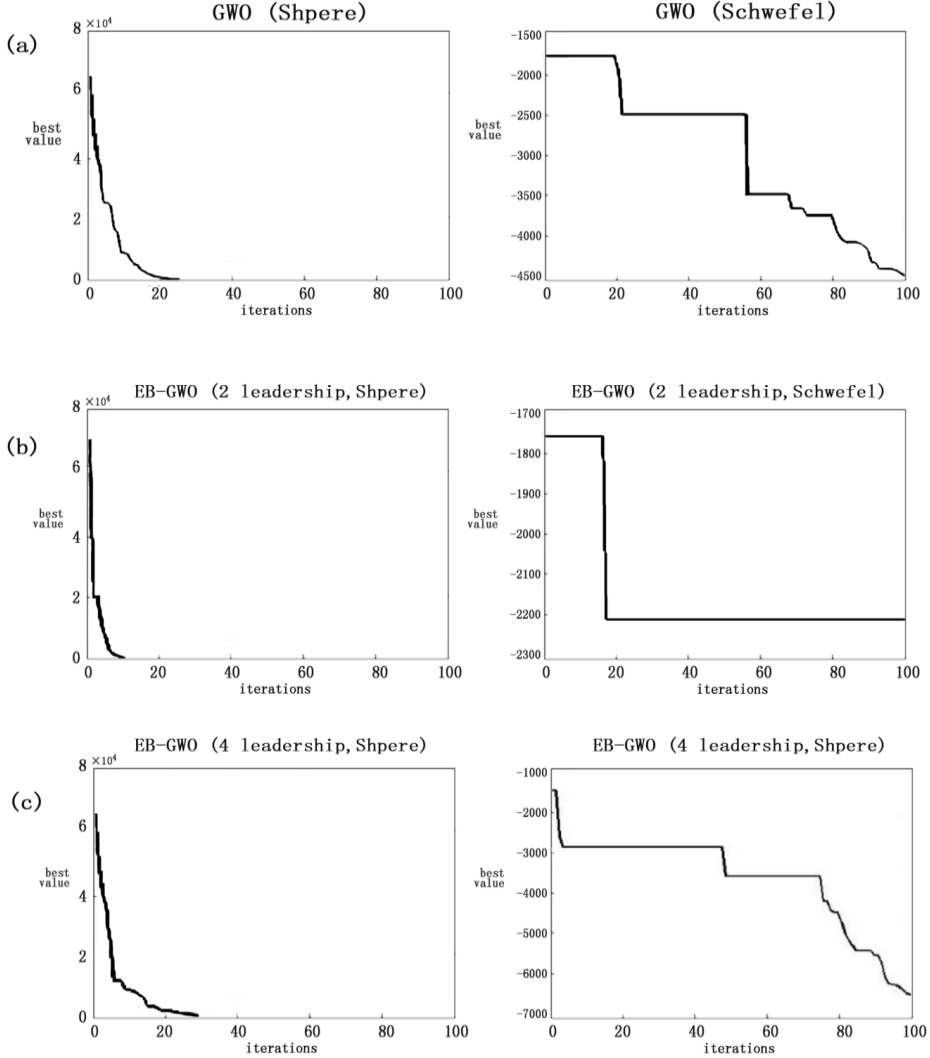


Figure 4: Tests of Best Value on Shere and Schwefel Functions.

When leaders wolves number is 4 (Figure 4 row (c)), it is similar to leaders wolves number is 3 (Figure 4 row (a)), its tracks were longer and larger, so they had greater exploration capabilities. Compared with Figure 4 row (a) and row(c), EBGWO's exploration capabilities are superior as compared to GWO. If the total algorithm's iterations numbers is extended beyond hundred times, it will not fall into the local optimal, but the needed very long time, thus the number of iterations can only be chosen based on the current scenario.

Experimental Results of China Credit Dataset and Analysis

To verify the comprehensive EBGWO algorithm performance, China_Credit dataset will be used in this section to test, the GWO's algorithm model trained and EBGWO algorithm will be evaluated. The dataset used is the China_Credit dataset, the algorithm running time is in Table 7.

Table 7. Algorithm Running Time.

Algorithm	Time
GWO	6.76s
EBGWO	4.33s

It can be seen from the experiments in section 4.2.2 that the optimal values of GWO and EBGWO algorithms tend to be stable around 100 iterations. Therefore, in the test of the China_Credit dataset, 100 iterations are taken as the baseline, and then 20, 50, 80, 100, 120, 150, 180 iterations are carried out. Here are the results shown in Table 8.

Table 8. Model Validation Test Results.

Algorithm	20 (iteration)	50 (iteration)	80 (iteration)	100 (base line)	120 (iteration)	150 (iteration)	180 (iteration)
Name	Accuracy /AUC	Accuracy /AUC	Accuracy /AUC	Accuracy /AUC	Accuracy /AUC	Accuracy /AUC	Accuracy /AUC
GWO	69.52/69.01	70.73/69.89	71.81/70.85	72.50/71.39	72.80/72.01	72.79/72.21	72.98/72.14
EBGWO	70.56/69.97	71.90/71.36	72.45/71.22	72.41/71.26	72.50/72.09	72.43/71.88	72.49/71.95

Table 7 shows that the test time of the improved algorithm of EBGWO in the China_Credit dataset was less than that of the GWO algorithm, which also confirms that the EBGWO algorithm can adjust the number of leader wolves according to the actual situation, thus improving the efficiency of the algorithm.

As can be seen from Table 8, with 100 iterations as the baseline, GWO tends to be stable (optimal solution) only after 100 iterations, while the EBGWO algorithm tends to be stable about 80 iterations, which also indicates that the EBGWO algorithm is more efficient in finding parameters than the GWO algorithm.

CONCLUSION

In the experimental analysis, the conclusion is leaders' number wolves of GWO is fixed at 3, and the test analysis in 20 benchmarking functions shows that the local optimal solution is easy to fall because it cannot timely adjust the number of leaders wolves according to the actual situation. The number of leaders wolves in the modified EBGWO algorithm is not fixed but selected according to the actual leaders wolves number, which makes the algorithm converge quickly and avoids embedding the algorithm into the local optimal solution. At the same time, this feature is also shown in the test of the risk loan dataset. When type of data is between good and bad, the algorithm chooses to leaders the wolves number to be 2. At this time, the development ability of the algorithm is strong (attack prey stage), which can make the algorithm converge quickly. The data between good and bad needs to increase the number of leading wolves (4 leading wolves), so that the algorithm is further in the search stage so that the leaders wolves can choose more accurate positions, so

as to improve the probability of successful hunting of wolves, that is, you can better distinguish these data between good and bad.

CONCLUSION AND PROSPECT

- i. Compared with the GWO algorithm that fixed the number of leading wolves, the EBGWO that could adjust the number of leading wolves had a better effect. The adaptability of EBGWO algorithm shows strong parameter searching ability in risk control model.
- ii. EBGWO algorithm can avoid the problem of easy local convergence of grey wolf algorithm (GWO) to a certain extent by improving the example learning mode, thereby improving work efficiency.

REFERENCES

- [1] R. Ahmed, G. P. Rangaiah, S. Mahadzir, S. Mirjalili, M. H. Hassan, S. Kamel, *Memory, evolutionary operator, and local search based improved Grey Wolf Optimizer with linear population size reduction technique*, Knowl.-Based Syst. **264** (2023), 110297.
- [2] A. Gupta, G. Gupta, *Comparative Study of Random Forest and Neural Network for Prediction in Direct Marketing*, in: H. Malik, S. Srivastava, Y.R. Sood, A. Ahmad (Eds.) *Applications of Artificial Intelligence Techniques in Engineering*, Springer Singapore, 2019, 401-410.
- [3] N. M. Hatta, A. M. Zain, R. Sallehuddin, Z. Shayfull, Y. Yusoff, *Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014–2017)*, Artif. Intell. Rev. **52** (2019), 2651-2683.
- [4] A. A. Heidari, R. Ali Abbaspour, H. Chen, *Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training*, Applied Soft Computing **81** (2019), 105521.
- [5] T. Huang, J. Lv, C. Sun, A. V. Tuzikov, *Advances in Neural Networks–ISNN 2018: 15th International Symposium on Neural Networks, ISNN 2018, Minsk, Belarus, June 25–28, 2018, Proceedings*, Springer, 2018.
- [6] V. K. Kamboj, *A novel hybrid PSO–GWO approach for unit commitment problem*, Neural Comput. Appl. **27** (2016), 1643-1655.
- [7] B. Mahdad, K. Srairi, *Blackout risk prevention in a smart grid based flexible optimal strategy using Grey Wolf-pattern search algorithms*, Energy Conversion and Management **98** (2015), 411-429.
- [8] S. Mirjalili, S. M. Mirjalili, A. Lewis, *Grey Wolf Optimizer*, Adv. Eng. Softw. **69** (2014), 46-61.
- [9] J. Nasiri, F. M. Khiyabani, *A whale optimization algorithm (WOA) approach for clustering*, Cogent Math. Stat. **5** (2018), 1483565.
- [10] R. Nisbet, J. Elder, G. Miner, *Accessory Tools for Doing Data Mining*, in: R. Nisbet, J. Elder, G. Miner (Eds.) *Handbook of Statistical Analysis and Data Mining Applications*, Academic Press, 2009, 99-117.
- [11] K. Sambrekar, V. S. Rajpurohit, S. Bhagyashri Deyannavar, *Design of a Cloud-Based Framework (SaaS) for Providing Updates on Agricultural Products*, in: M.S. Reddy, K. Viswanath, S.P. K.M (Eds.) *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, Springer Singapore, 2018, 115-122.
- [12] N. Singh, S. B. Singh, *Hybrid Algorithm of Particle Swarm Optimization and*

- Grey Wolf Optimizer for Improving Convergence Performance*, J. Appl. Math. **2017** (2017), 2030489.
- [13] G. Wu, R. Mallipeddi, P. N. Suganthan, *Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization*, Nanyang Technological University, 2017.
- [14] X. Yao, Y. Liu, G. Lin, *Evolutionary programming made faster*, IEEE Transactions on Evolutionary Computation **3** (1999), 82-102.
- [15] S. Zhang, Y. Zhou, *Grey Wolf Optimizer Based on Powell Local Optimization Method for Clustering Analysis*, Discrete Dyn. Nat. Soc. **2015** (2015), 481360.