Enhanced Gold Rush Optimizer for Feature Selection: Application in Software Fault Prediction Datasets

Yaxi Qin and Jian Zhao

Abstract—**The Gold Rush Optimizer (GRO) is a popular metaheuristic algorithm proposed recently. However, it has slow convergence, low accuracy, and easily gets stuck in local optima when solving real problems. To address these issues, we propose an enhanced version called the Enhanced Gold Rush Optimizer (EGRO). Our algorithm achieves better performance through four new mechanisms. First, a Sinusoidal Bridging Mechanism uses the sine function's periodic waves to boost global search. Second, we create a new partner selection strategy based on Euclidean distance. Third, an adaptive Levy flight strategy dynamically adjusts the search step size and direction to improve population diversity. Fourth, a Metal Detector Strategy combines gradient feedback of the "M" factor to accurately avoid local optima traps. To test EGRO, we build a two-level evaluation system. Tests on the CEC2022 benchmark sets show that EGRO performs best in convergence accuracy and stability for unconstrained optimization. Meanwhile, in the feature selection task on 16 software fault prediction datasets, EGRO outperforms mainstream optimization algorithms by improving classification accuracy by 0.1%-4.5%. It achieves the minimum number of features in 87.5% of the datasets and obtains the optimal fitness value in 93.75% of the datasets. Experiments prove that EGRO has strong scalability and practical value.**

Key Words—**Gold rush optimizer, Sinusoidal bridging mechanism, Feature selection, Software fault prediction.**

I. INTRODUCTION

DATA is growing very fast in both types and amounts. This makes it hard to find and check useful information from big data. Researchers working in data mining are actively trying to improve methods for sorting data [1, 2] and developing machine learning techniques [3, 4]. Many data mining tasks deal with huge datasets containing many features, but often only some features really matter. The extra features become like noise, unimportant, or repeated [5]. Choosing a smaller set of features that truly represents the whole dataset greatly affects how well machine learning works. This impact both the accuracy of predictions and how much time the calculations take . [6].

Manuscript received June 30, 2025; revised July 8 and July 16, 2025; accepted July 24, 2025. This article was recommended for publication by Associate Editor Shujin Qin upon evaluation of the reviewers' comments.

Y. Qing is with the School of Mathematics and Physics, University of Science and Technology Liaoning, Anshan 114051, China (e-mail: xyla@ustl.edu.cn).

J. Zhao is with the School of Science, University of Science and Technology Liaoning, Anshan 114051, China (e-mail: zhao@ustl.edu.cn).

Corresponding Author: Jian Zhao

Feature selection plays a vital role in data mining and machine learning. By eliminating redundant and irrelevant features, FS identifies optimal feature subsets that improve model performance while reducing computational overhead [7]. This dual benefit of enhanced efficiency and reduced resource consumption makes FS indispensable across diverse fields – from bioinformatics [8] to computer vision applications like image search [9] and recognition [10], as well as text analysis tasks including mining [11], classification [12], and image categorization [13].There are three common ways to do feature selection: filter-based FS, wrapper-based FS, and embedded FS [14]. Filter-based FS uses stats to give each feature in a dataset a score. It does not look at how features depend on each other [15]. Then it ranks features by their scores, removes the low-ranked ones, and keeps the high-ranked ones [16]. Well-known filter-based FS methods are document frequency [17], information gain [18], and chi-square [19]. Wrapper-based FS uses search methods like particle swarm optimization (PSO) [20] or genetic algorithms (GA) [21] to test different groups of features. After the search, it uses classifiers like decision trees [22], naive Bayes [23], or k-nearest neighbors [24] to check how good the selected features are. Embedded FS mixes parts of both filter-based and wrapper-based methods. It puts search methods inside a classifier so that the classifier can pick features that make it accurate [25]

Feature selection problems belong to the NP-hard category, meaning finding optimal solutions quickly is extremely challenging [26]. This makes optimization methods crucial for discovering satisfactory solutions efficiently. Recently, meta-heuristic algorithms have gotten a lot of attention in the research world for solving optimization problems.

We choose GRO [27] as one of the new meta-heuristic algorithms to talk about the benefits of using meta-heuristic algorithms for FS problems. GRO is a new algorithm inspired by human behavior. It is easy to use, has a simple structure, and needs few parameters to control. It also has good stability and finds solutions well. This study aims to leverage these advantages for optimal feature selection [28]. Many optimization problems, including FS, involve binary search spaces and decision variables. Furthermore, the group update mechanism in GRO can alter the types of individuals within the population.

The proposed GRO has many benefits for solving feature selection optimization problems. First, it can adapt to different types and levels of difficulty in feature selection. This flexibility is particularly valuable, as many FS problems are inherently complex and benefit from an approach requiring minimal parameter tuning, a key characteristic of GRO. Second, GRO has a simple design. It finds overall solutions quickly and accurately, offering high convergence rates for tough FS problems. The goal of this work is to create a wrapper-based method using an improved GRO algorithm to solve many failure prediction and classification problems. This depends on picking the most important and useful attributes from specific datasets. These are needed to build the best classification models with high performance, fewer features, and shorter run times.

To achieve the goal of this study, the following four contributions were made:

1) Introduced a Sinusoidal bridging mechanism to optimize l_e, improving the global exploration ability of GRO and increasing the algorithm's convergence speed.
2) Introduced a new partner selection strategy based on Euclidean distance. It expands the algorithm's search space and helps avoid getting trapped in local optima too early.
3) Introduced a gold panning factor M to more reasonably distribute the different action stages of the gold panners, speeding up algorithm convergence and improving accuracy.
4) Introduced a metal detector strategy combined with the M factor to help gold panners more effectively locate gold mines, speeding up the algorithm's convergence.

The rest of this paper is structured as follows: Section II introduces the basic GRO algorithm. Section III provides a detailed explanation of the proposed method. Sections IV to V describe the experimental setup, the obtained results, and the comprehensive analysis. Finally, the paper concludes with research conclusions and future recommendations.

II. Gold Rush Optimizer

A. GRO Mathematical Model and Algorithm

The Gold Rush Optimization (GRO) algorithm is proposed by Kamran Zolf in 2023. It is a metaheuristic optimization algorithm inspired by the Gold Rush activities. It simulates the migration, cooperation, and gold panning behaviors of prospectors during the Gold Rush. This helps achieve an optimized search process.The migration process of the gold panners is represented by equations (1) and (2).

$$\overrightarrow{D_1} = \overrightarrow{C_1} * \overrightarrow{X^*(t)} - \overrightarrow{X_t(t)} \tag{1}$$

$$\overrightarrow{Xnew_t}(t+1) = \overrightarrow{X_t(t)} + A_1 * \overrightarrow{D_1} \tag{2}$$

where, $X^*(t)$ and $X_i(t)$ represent the positions of the best gold mine and the i-th gold panner, respectively. is the current iteration number, and $Xnew_i(t+1)$ is the new position of the gold panner. A_1 and C_1 are vector coefficients calculated using equations (3) and (4).

$$A_1 = 1 + l_1 * \left(\overrightarrow{r_1} - \frac{1}{2}\right) \tag{3}$$

$$\overrightarrow{C_1} = 2\overrightarrow{r_2} \tag{4}$$

r_1 and r_2 are random vectors with values in the range [0,1]. l_1 and l_2 are the convergence components defined by equation (5). For values greater than 1, they decrease non-linearly. Here, max_{iter} is the maximum number of iterations, and iter is the current iteration number.

$$l_e = \left(\frac{\text{max}_{\text{iter}} - \text{iter}}{\text{max}_{\text{iter}} - 1}\right)^e \left(2 - \frac{1}{\text{max}_{\text{iter}}}\right) + \frac{1}{\text{max}_{\text{iter}}} \tag{5}$$

where, e is equal to 1 or 2.The formula for the gold mining stage is:

$$\overrightarrow{D_2} = \overrightarrow{X_l}(t) - \overrightarrow{X_r}(t) \tag{6}$$

$$\overrightarrow{Xnew_l}(t+1) = \overrightarrow{X_r}(t) + A_2 * \overrightarrow{D_2} \tag{7}$$

where, $X_r(t)$ represents the position of the randomly selected gold panner. A_2 is calculated using equation (8).

$$A_2 = 2l_2\vec{r_1} - l_2 \tag{8}$$

Gold panning is done through teamwork. This stage is represented by equations (9) and (10). In these equations, $X_{g1}(t)$ and $X_{g2}(t)$ are two randomly selected gold panners. D_3 is the cooperation vector.

$$\overrightarrow{Xnew_l}(t+1) = \overrightarrow{X_l}(t) + \vec{r_1} * \overrightarrow{D_3} \tag{9}$$

$$\overrightarrow{D_3} = \overrightarrow{X_{g2}}(t) - \overrightarrow{X_{g1}}(t) \tag{10}$$

Gold prospectors continuously evaluate their position. A key parameter in their decision is to get more gold. To decide whether to stay at the current position or move to a new one, they compare the two positions using a fitness function. If the value of the objective function improves, the prospector updates their position. Otherwise, they stay at the previous position. This position is modeled by equation (11) in the minimization problem:

$$\overrightarrow{X_l}(t+1) = \overrightarrow{Xnew_l}(t+1) \quad \text{if} \quad f(\overrightarrow{Xnew_l}(t+1)) < f(\overrightarrow{X_l}(t)) \tag{11}$$

III. Proposed Enhancement Methods of GRO

This section presents four novel improvement strategies.

A. Sinusoidal bridging mechanism

We see positive effects from using alternative bridging mechanisms in our methods. In these methods, experiments are done to change the known linear mechanisms. Inspired by the successful use of different math and chaotic functions, we suggest changes to the known GRO linear mechanism.This mechanism is governed by l_1 and l_2. The proposed changes are given by the following equations.

$$l_1 = 2 * \left(1 - \left(\frac{\sin\theta}{2}\right)^2\right) \tag{12}$$

$$l_2 = 2 * \left(1 - \left(\frac{\cos\theta}{2}\right)^2\right) \tag{13}$$

$$\theta = \frac{\pi * \text{iter}}{\text{max}_{\text{iter}}} \tag{14}$$

B. Partner selection strategy

In the original GRO, gold panners start by randomly choosing two partners. This means new solutions are created without considering the structure of the partners. Many classic algorithms build new solutions by using the structural relationships of the current generation. Thus, how to fully use the structural information of partners to achieve better cooperation is a key question.

The collaboration process on the left side of Figure 1 shows the defects when partners are randomly selected. When miners choose partners closer to the optimal solution on the same side, the updated position area of the miners appears as a square region within the blue area, rather than being closer to the optimal solution. In this case, many poor solutions are generated, resulting in slow convergence speed and low accuracy. Therefore, a new method is proposed to overcome this limitation.

First, the number of partners is controlled by $Pp = 0.5$, and the population is divided into gold panners and partners based on fitness values. The Euclidean distance between two individuals, $X_i = (x_{(i,1)}, x_{(i,2)}, x_{(i,3)}, \ldots, x_{(i,D)})$ and $X_j = (x_{(j,1)}, x_{(j,2)}, x_{(j,3)}, \ldots, x_{(j,D)})$, is calculated using equation(15) to measure their similarity. To avoid high similarity in the population, partners with the smallest similarity are selected for cooperation. Once a partner is chosen, it cannot be reused.

$$d_{ij} = \left(\sum_{k=1}^{D} (x_{i,k} - x_{j,k})^2\right)^{1/2} \tag{15}$$

As shown in the collaboration process on the right side of Figure 1, the farthest partners are selected for cooperation. This allows the gold mine to appear in the blue square area containing the optimal solution. As a result, gold panners can quickly converge near the optimal solution instead of searching blindly.

C. Adaptive Lévy Flight Strategy

Lévy flight is a random walk model. The chance of taking a step follows a heavy-tailed distribution. It mixes long and short walks. This speeds up how fast the algorithm finds solutions. To make the algorithm more accurate, we use a weighted Lévy flight.

The calculation of the Lévy flight distribution function is as follows:

$$Levy(D) = s * \frac{\mu * \sigma}{|\nu|^{\frac{1}{\eta}}} \tag{16}$$

where, s is a fixed constant of 0.01, and η is a fixed constant of 1.5. μ and ν are random numbers in the interval [0,1]. The formula for σ is as follows:

$$\sigma = \left(\frac{\Gamma(1+\eta) * \sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{1+\eta}{2}\right) * \eta * 2^{\left(\frac{\eta-1}{2}\right)}}\right)^{\frac{1}{\eta}} \tag{17}$$

where, Γ represents the gamma function, and the value of η is 1.This paper uses Lévy flight to optimize the migration vector D_4

$$\overrightarrow{D_4} = \overrightarrow{C_1} * \overrightarrow{X^*}(t) * Levy - \overrightarrow{X_l}(t) \tag{18}$$

D. Metal detector strategy

In the GRO algorithm, the three strategies of migration, gold panning, and cooperation are evenly shared based on a random number m. This paper designs a gold panning factor M based on the BMI index of adult males, the average walking speed of adults, and the distance between gold panners and gold mines, as shown in the following formula:

$$M = \frac{1}{R} * \left(1 + \frac{\text{iter}}{\max_{\text{iter}}}\right) \tag{19}$$

the expression for R is as follows:

$$R = \frac{B * V^2}{L} \tag{20}$$

where B is the BMI index of adult men, a random number between [18.5, 23.9]. V is the average walking speed of adults, set at 4.5 km/h. L is the distance between the prospector and the gold mine, a random number between [0, 100].The design of the gold panning factor is inspired by the impact of adult men's physical fitness and movement speed on their ability to successfully reach the target gold mine during the gold rush. Since there are no clear historical records for these data, we use the BMI index of adult men as a quantitative measure of physical fitness and the average walking speed of adults as the movement speed of the gold panners. Theoretically, the distance between the gold panner and the gold mine will not exceed 100 km. M is a random number between [0, 0.4].

In the original algorithm, the three strategies were optimized with equal one-third probabilities, which did not effectively balance exploration and exploitation. Therefore, EGRO achieves better equilibrium between exploration and exploitation phases by utilizing parameter M: When $M > 0.2$, it is the collaboration phase (exploitation phase), and at all other times, it corresponds to the migration and mining phase (exploration phase).

In the gold panning process, a gold panning factor helps balance the algorithm's exploitation and exploration phases. The migration and cooperation steps are the exploration phases, while the gold mining step is the exploitation phase. The migration step performs medium exploration using random moves to search widely. The cooperation step performs strong exploration using long Lévy flight jumps. To better balance exploitation and exploration in EGRO, we update positions using this equation:

$$\begin{cases} \overrightarrow{Xnew_l}(t+1) = \overrightarrow{X_r}(t) + A_2 * \sin(r_1) * \overrightarrow{D_2} & , \ M > 0.2 \\ \overrightarrow{Xnew_l}(t+1) = \overrightarrow{X^*}(t) + A_1 * \cos(r_2) * \overrightarrow{D_3} & , \ M > 0.05 \\ \overrightarrow{Xnew_l}(t+1) = \overrightarrow{X_l}(t) + A_1 * \overrightarrow{D_4} & , \ \text{else} \end{cases} \tag{21}$$

where, r_1 and r_2 are random numbers between [0, 1].

As shown in Figure 1, the original algorithm (left) randomly selects two collaborators for cooperative strategies. While this ensures randomness, it may limit the current individual's ability to efficiently locate the gold mine. To tackle this issue, we propose an enhanced partner selection method: First, we divide the population into two equal groups. From the first group, we randomly select a collaborator X_{g1}, then choose the individual

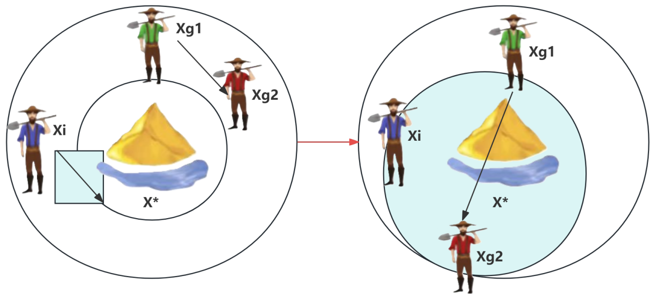

Fig. 1. Partner selection strategy.

Algorithm 1 Enhanced Gold Rush Optimization

Input: population size N, maximum iterations $\max_{\text{iter}}$
Output: the best solution X^*
Begin
Initialize the gold prospector's population X_i, $i = 1, 2, \ldots, N$

Initialize the gold prospector's new population $Xnew_i = X_i$, $i = 1, 2, \ldots, N$
Initialize t, l_1, l_2, M
X^* is the best search agent
while $t \leqslant \max_{\text{iter}}$ **do**
  for $i = 1$ **to** N **do**
    Calculate the fitness of $Xnew_i$
    Update X_i according to equation(21)
    Update X^*
  end for
  Update l_1, l_2 by equations(12)(13)
  Calculate M by equation(19)
  for $i = 1$ **to** N **do**
    Update $Xnew_i$ using migration, mining or collaboration
  end for
  $t \leftarrow t + 1$
end while
return X^*
End

in the second group that has the farthest Euclidean distance from X_{g1} as the second collaborator X_{g2}. This strategic pairing significantly expands the search scope while accelerating gold mine localization.

IV. EGRO QUANTITATIVE ANALYSIS

In this section, we compare EGRO, GRO, and seven other popular algorithms on the CEC2022 test suites to verify the performance of EGRO.

A. Algorithm parameter settings

In this section, to verify the effectiveness of EGRO, we compare it with eight advanced algorithms on the CEC 2022 test suite. The compared algorithms include: Fata Morgana Algorithm(FATA) [29], Differential Evolution Algorithm(DE) [30], Grey Wolf Optimizer (GWO) [31], Whale Optimization Algorithm(WOA)[32], Catch Fish Optimization Algorithm(CFOA) [33], Black-winged Kite Algorithm(BKA) [34], Secretary Bird Optimization Algorithm(SBOA) [35] and GRO [27].

The overall size for all algorithms is set to 30, with 500 iterations. Each algorithm runs independently 30 times, and the relevant results are recorded. For each test function and its corresponding dimension, the best result is highlighted in bold.

B. CEC 2022 EXPERIMENTAL RESULTS

To further verify the scalability of EGRO, in this section, we test it using the CEC2022 benchmark functions and compare it with eight other popular algorithms. The CEC2022 benchmark functions include unimodal, multimodal, hybrid, and composite functions.

The statistical results of EGRO, GRO, and 7 other comparison algorithms on the 10-dimensional and 20-dimensional CEC2022 benchmark functions are shown in Tables I and II .

As shown in Tables I and II, in the 10-dimensional search space, EGRO achieves a good balance between global exploration and local exploitation. Compared to traditional metaheuristic algorithms such as FATA, DE, and GWO, EGRO demonstrates superior convergence accuracy, more stable convergence, and better performance in terms of mean and optimal value indicators for most test functions. This clearly shows that EGRO, with its unique search mechanisms, effectively avoids local optima traps and accurately identifies the global optimum when handling low-dimensional complex multimodal optimization problems. When the optimization space expands to a 20-dimensional high-dimensional scenario, the algorithm's complexity and search difficulty increase sharply. Nevertheless, EGRO still shows strong environmental adaptability and effectively maintains search efficiency and accuracy in high-dimensional spaces. This suggests that EGRO can more efficiently explore potential solution areas in high-dimensional solution spaces while balancing the exploration-exploitation process.

In general, compared to other metaheuristic algorithms, EGRO's advantages in the CEC 2022 10-dimensional and 20-dimensional test scenarios can be systematically summarized into three points: First, its exceptional convergence accuracy: In multi-function test tasks, the mean and optimal value indicators consistently exhibit characteristics that closely approach the theoretical optimal solution, highlighting a significant advantage in solution quality.Second, its ability to generalize across dimensions: During the testing process from low-dimensional to high-dimensional spaces, the algorithm's performance degradation is significantly smaller compared to the comparison algorithms, validating its strong generalization capability in complex optimization tasks across varying dimensions.Third, the efficiency of its search mechanism: By improving the population evolution logic and information utilization mode of the metaheuristic algorithm, EGRO achieves an efficient collaboration between global exploration and local refinement search, providing a more competitive solving paradigm for complex dimensional optimization problems, showcasing its potential advantages and application value in handling multi-dimensional complex optimization tasks within the metaheuristic algorithm system.

TABLE I CEC 2022 Experimental results (Dim=10)

ID	Index	FATA	DE	GWO	WOA	CFOA	BKA	SBOA	GRO	EGRO
	Mean	2.18E+03	8.31E+03	2.84E+03	2.69E+04	5.39E+02	7.35E+02	5.24E+02	4.17E+02	**3.00E+02**
F1	Std	1.76E+03	2.89E+03	2.56E+03	1.02E+04	2.06E-02	1.16E+03	2.34E+02	1.80E+02	2.60E+03
	Best	4.14E+02	4.85E+03	3.73E+02	8.88E+03	3.13E+02	3.03E+02	3.49E+02	3.17E+02	3.00E+02
	Mean	4.25E+02	4.11E+02	4.14E+02	4.52E+02	4.03E+02	4.30E+02	4.17E+02	**4.02E+02**	4.04E+02
F2	Std	2.87E+01	8.62E+01	1.40E+01	6.85E+01	3.46E+01	4.07E+01	2.58E+01	2.89E+01	3.96E+00
	Best	4.00E+02	4.01E+02	4.03E+02	4.01E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02	4.00E+02
	Mean	6.22E+02	**6.00E+02**	6.01E+02	6.44E+02	6.02E+02	6.31E+02	6.00E+02	6.01E+02	6.00E+02
F3	Std	7.85E+00	6.41E-04	1.55E+00	1.61E+01	1.67E+01	1.23E+01	2.33E-01	9.17E-01	9.20E-03
	Best	6.08E+02	6.00E+02	6.00E+02	6.12E+02	6.00E+02	6.08E+02	6.00E+02	6.00E+02	6.00E+02
	Mean	8.28E+02	8.24E+02	8.16E+02	8.40E+02	**8.11E+02**	8.20E+02	8.14E+02	8.12E+02	8.12E+02
F4	Std	5.96E+00	5.03E+00	7.15E+00	1.93E+01	3.37E+00	5.64E+00	4.36E+00	5.21E+00	4.19E+00
	Best	8.12E+02	8.14E+02	8.05E+02	8.14E+02	8.05E+02	8.10E+02	8.05E+02	8.04E+02	8.05E+02
	Mean	1.18E+03	9.37E+02	9.12E+02	1.65E+03	9.01E+02	1.14E+03	9.02E+02	9.04E+02	**9.00E+02**
F5	Std	2.32E+02	2.49E+01	1.79E+01	4.43E+02	9.26E-01	1.38E+02	7.01E+00	7.92E+00	5.29E-01
	Best	9.03E+02	9.07E+02	9.00E+02	1.04E+03	9.00E+02	9.11E+02	9.00E+02	9.00E+02	9.00E+02
	Mean	6.01E+03	8.61E+03	6.02E+03	5.83E+03	3.02E+03	3.21E+03	4.09E+03	**2.73E+03**	3.22E+03
F6	Std	4.45E+03	5.72E+03	2.31E+03	3.66E+03	1.13E+03	1.62E+03	2.20E+03	1.01E+03	1.48E+03
	Best	1.99E+03	2.31E+03	2.38E+03	2.02E+03	1.87E+03	1.89E+03	2.00E+03	1.84E+03	1.85E+03
	Mean	2.04E+03	**2.01E+03**	2.03E+03	2.08E+03	2.03E+03	2.05E+03	2.02E+03	2.02E+03	2.01E+03
F7	Std	1.95E+01	3.37E+00	1.14E+01	2.82E+01	6.53E+00	2.25E+01	7.69E+00	8.21E+00	9.77E+00
	Best	2.01E+03	2.00E+03	2.01E+03	2.04E+03	2.01E+03	2.02E+03	2.00E+03	2.00E+03	2.00E+03
	Mean	2.33E+03	2.22E+03	2.23E+03	2.34E+03	2.22E+03	2.32E+03	2.22E+03	2.22E+03	**2.22E+03**
F8	Std	4.47E+00	3.90E+00	2.35E+01	9.65E+00	5.65E+00	2.02E+00	5.25E+00	4.10E+00	7.96E+00
	Best	2.21E+03	2.21E+03	2.21E+03	2.23E+03	2.21E+03	2.21E+03	2.20E+03	2.20E+03	2.20E+03
	Mean	2.55E+03	2.53E+03	2.57E+03	2.61E+03	2.53E+03	2.58E+03	2.53E+03	2.53E+03	**2.53E+03**
F9	Std	2.16E+01	1.68E+00	3.24E+01	5.01E+01	8.89E+01	3.35E+01	6.15E+01	2.50E-01	1.41E-11
	Best	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03	2.53E+03
	Mean	2.58E+03	**2.49E+03**	2.62E+03	2.61E+03	2.53E+03	2.63E+03	2.56E+03	2.55E+03	2.51E+03
F10	Std	6.48E+01	1.86E+01	2.07E+02	1.72E+02	4.92E+01	1.64E+02	5.99E+01	5.64E+01	2.80E+01
	Best	2.50E+03	2.43E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03
	Mean	2.78E+03	2.74E+03	2.91E+03	2.86E+03	2.64E+03	2.80E+03	2.71E+03	2.66E+03	**2.61E+03**
F11	Std	1.61E+02	2.85E+01	1.72E+02	1.69E+02	8.86E+01	2.80E+02	1.74E+02	9.78E+01	3.82E+01
	Best	2.61E+03	2.64E+03	2.72E+03	2.67E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03
	Mean	2.88E+03	2.87E+03	2.87E+03	2.90E+03	**2.86E+03**	2.87E+03	2.86E+03	2.87E+03	2.87E+03
F12	Std	1.46E+01	7.72E-01	1.10E+01	3.41E+01	1.49E+00	1.45E+01	1.03E+00	2.55E+00	1.41E+00
	Best	2.86E+03	2.86E+03	2.86E+03	2.87E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03	2.86E+03

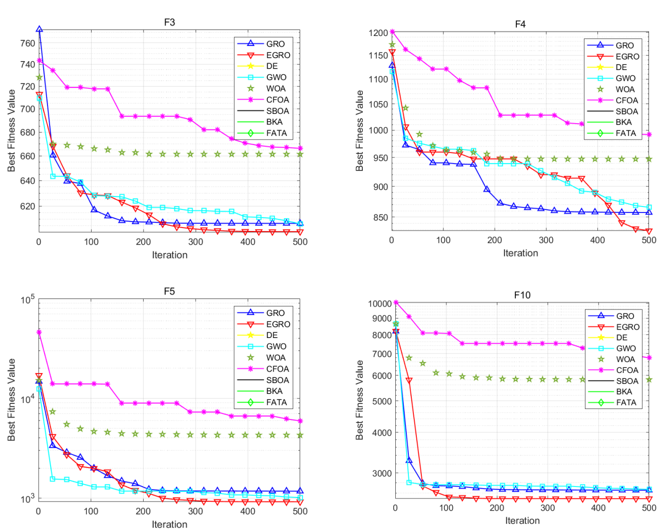

Fig. 2. CEC2022 test function comparison curve (Dim=20).

As shown in the convergence curve in Figure 2, EGRO demonstrates both fast convergence and high accuracy. During the optimization process, EGRO performs well, especially in the early and middle stages, where it quickly identifies better solutions. In the later stages, it maintains a high solution quality.

V. APPLICATION OF EGRO

To evaluate EGRO's performance in solving real-world optimization problems, further testing is conducted using a feature selection problem based on software fault prediction datasets. EGRO is compared with seven other algorithms.

In this section, the effectiveness, robustness, and stability of the proposed algorithm are studied using 16 feature selection datasets related to software fault prediction. The characteristics of the datasets are described in section 5.1. The parameter settings for the proposed algorithm are detailed in section 5.2. Section 5.3 explains the evaluation methods used to assess the performance of the proposed algorithm. Section 5.4 demonstrates the impact of the suggested improvements on the GRO algorithm's performance and compares the performance

TABLE II CEC 2022 Experimental results (Dim=20)

ID	Index	FATA	DE	GWO	WOA	CFOA	BKA	SBOA	GRO	EGRO
F1	Mean	1.34E+04	3.51E+04	1.58E+04	3.58E+04	1.30E+04	1.18E+04	6.44E+03	1.28E+04	**2.77E+03**
	Std	4.45E+03	6.62E+03	5.29E+03	1.15E+04	5.91E+03	6.39E+03	2.62E+03	3.49E+03	2.14E+03
	Best	5.76E+03	2.63E+04	5.90E+03	2.05E+04	3.86E+03	3.42E+03	2.11E+03	7.23E+03	6.18E+02
F2	Mean	5.51E+02	4.58E+02	4.99E+02	6.43E+02	5.04E+02	6.87E+02	4.67E+02	5.00E+02	**4.56E+02**
	Std	5.58E+01	6.21E+00	4.58E+01	7.48E+01	3.55E+01	2.62E+02	2.13E+01	3.03E+01	1.13E+01
	Best	4.60E+02	4.49E+02	4.52E+02	4.87E+02	4.54E+02	4.84E+02	4.35E+02	4.69E+02	4.45E+02
F3	Mean	6.56E+02	**6.00E+02**	6.08E+02	6.71E+02	6.16E+02	6.55E+02	6.01E+02	6.09E+02	6.01E+02
	Std	8.18E+00	1.09E-02	4.64E+00	1.53E+01	7.03E+00	8.93E+00	1.63E+00	3.62E+00	6.41E-01
	Best	6.36E+02	6.00E+02	6.02E+02	6.45E+02	6.06E+02	6.34E+02	6.00E+02	6.03E+02	6.00E+02
F4	Mean	9.06E+02	9.12E+02	8.59E+02	9.40E+02	8.62E+02	8.87E+02	8.46E+02	**8.44E+02**	8.45E+02
	Std	8.43E+00	1.11E+01	2.35E+01	2.72E+01	1.29E+01	1.67E+01	1.28E+01	1.03E+01	3.41E+01
	Best	8.94E+02	8.95E+02	8.26E+02	9.00E+02	8.38E+02	8.54E+02	8.26E+02	8.25E+02	8.18E+02
F5	Mean	2.69E+03	1.74E+03	1.30E+03	4.33E+03	1.10E+03	2.30E+03	9.98E+02	1.07E+03	**9.33E+02**
	Std	1.81E+02	2.88E+02	2.74E+02	1.90E+03	1.47E+02	4.21E+02	1.15E+02	1.69E+02	2.86E+01
	Best	2.14E+03	1.17E+03	9.72E+02	2.05E+03	9.24E+02	1.39E+03	9.05E+02	9.33E+02	9.07E+02
F6	Mean	3.32E+05	2.10E+06	5.71E+06	8.68E+06	**3.52E+03**	2.27E+06	1.21E+04	2.64E+04	8.21E+03
	Std	3.72E+05	1.13E+06	1.26E+07	1.80E+07	1.57E+03	1.23E+07	1.94E+04	3.43E+04	5.64E+03
	Best	6.86E+04	3.37E+05	5.84E+03	2.24E+05	1.94E+03	3.34E+03	2.08E+03	2.05E+03	1.88E+03
F7	Mean	2.16E+03	2.05E+03	2.11E+03	2.33E+03	2.09E+03	2.12E+03	2.05E+03	2.06E+03	**2.04E+03**
	Std	2.62E+01	1.10E+01	5.81E+01	5.74E+01	1.57E+01	2.97E+01	2.33E+01	2.07E+01	1.44E+01
	Best	2.12E+03	2.04E+03	2.04E+03	2.13E+03	2.07E+03	2.07E+03	2.03E+03	2.03E+03	2.02E+03
F8	Mean	2.28E+03	2.23E+03	2.25E+03	2.33E+03	2.23E+03	2.29E+03	2.23E+03	2.23E+03	**2.23E+03**
	Std	5.96E+01	1.56E+00	4.63E+01	8.26E+01	6.31E+01	8.24E+01	4.52E+01	2.16E+00	5.22E+00
	Best	2.23E+03	2.23E+03	2.23E+03	2.24E+03	2.23E+03	2.23E+03	2.22E+03	2.22E+03	2.22E+03
F9	Mean	2.54E+03	2.48E+03	2.53E+03	2.59E+03	2.50E+03	2.61E+03	2.48E+03	2.49E+03	**2.48E+03**
	Std	2.21E+01	5.06E-01	2.35E+01	5.01E+01	1.22E+01	1.70E+02	5.49E-01	4.12E+00	2.85E-02
	Best	2.50E+03	2.48E+03	2.49E+03	2.51E+03	2.49E+03	2.49E+03	2.48E+03	2.48E+03	2.48E+03
F10	Mean	3.78E+03	2.53E+03	3.51E+03	5.08E+03	2.72E+03	4.14E+03	2.70E+03	2.77E+03	**2.51E+03**
	Std	1.13E+03	5.54E+01	7.95E+02	1.17E+03	6.26E+02	1.11E+03	3.14E+02	4.95E+02	3.95E+01
	Best	2.53E+03	2.49E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03	2.50E+03
F11	Mean	3.21E+03	2.95E+03	3.62E+03	3.89E+03	3.06E+03	4.82E+03	2.95E+03	3.15E+03	**2.93E+03**
	Std	1.82E+02	6.85E+01	3.94E+02	5.55E+02	1.20E+02	1.18E+03	1.75E+02	1.60E+02	4.49E+01
	Best	2.97E+03	2.90E+03	2.99E+03	3.41E+03	2.73E+03	3.45E+03	2.60E+03	2.83E+03	2.90E+03
F12	Mean	3.17E+03	2.96E+03	2.98E+03	3.08E+03	2.98E+03	3.07E+03	**2.95E+03**	2.98E+03	2.95E+03
	Std	7.61E+01	5.40E+00	2.60E+01	7.54E+01	2.09E+01	8.89E+01	9.44E+01	1.60E+01	9.14E+01
	Best	3.05E+03	2.95E+03	2.95E+03	2.99E+03	2.95E+03	2.96E+03	2.94E+03	2.96E+03	2.94E+03

of the proposed algorithm with 7 established feature selection methods.

A. Datasets

This section presents the characteristics of 16 real-world datasets that are directly related to the software fault prediction problem. These datasets are commonly used in the literature to evaluate the effectiveness of feature selection methods. The datasets are from two public repositories: PROMISE and OpenML.

Table III shows the main characteristics of the datasets used in this study. It includes the common name, number of classes (# Classes), actual number of features (# Features), number of faults (# Faults). Seven of the datasets (KC2, KC3, CM1, MW1, PC1, PC2, and PC5) were established by the National Aeronautics and Space Administration (NASA). These datasets include data related to satellites, ground stations, and other simulated data. The following datasets (Ar1, Ar3, Ar4, Ar5, and Ar6) were sampled from embedded software developed in C language. All datasets have two classes, with the number of features ranging from 21 to 39. The number of faults in these datasets ranges from 8 to 1759, reflected in fault percentages ranging from 2.15% to 32.29%. As previously mentioned, the integration of the k-NN classifier in the proposed EGRO version significantly enhances its effectiveness in handling these datasets, a claim well-supported by prior literature in the feature selection (FS) field.

For the experimental problem, each dataset is divided into a training set and a test set. 80% of the instances in each dataset are randomly selected for training, and the remaining instances are used for testing.

B. Parameters Configuration

Table IV shows the parameter settings for the proposed EGRO algorithm. The algorithm uses the same parameter settings to ensure all algorithms run under the same conditions. To study the robustness and stability of these algorithms, each algorithm is run 20 times independently with different random seeds. The proposed algorithm and other comparison algorithms are designed using MATLAB R2023a in a Windows 10 64-bit environment.

TABLE III Characteristics of the sixteen SFP dataset

#	Dataset	Dimension	#Classes	#Features	#Faults
1	Ar1	121	2	29	8
2	Ar3	63	2	29	8
3	Ar4	107	2	29	20
4	Ar5	36	2	29	8
5	Ar6	101	2	29	15
6	CM1	498	2	22	49
7	JM1	7782	2	22	1759
8	KC1	2109	2	22	325
9	KC2	522	2	21	107
10	KC3	194	2	39	43
11	MC2	125	2	39	52
12	MW1	403	2	37	31
13	PC1	1109	2	37	76
14	PC2	5589	2	36	23
15	PC3	1563	2	38	140
16	PC4	1458	2	38	178

TABLE IV Settings of the parameters of the proposed EGRO

Parameter	Value
Population size	20
Number of iterations	100
Runs	20

C. Evaluation Measures

The proposed EGRO-based algorithm and the other comparison algorithms are evaluated using appropriate metrics. These evaluation measures are summarized as follows:

Accuracy: It measures the success rate of the classifier by calculating the percentage of true classes across all categories, as shown in equation (22).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

where, TP (True Positive) represents the ratio of positive classes predicted by the classification model.TN (True Negative) reflects the ratio of negative classes predicted by the classification model. FP (False Positive) represents the ratio of negative classes incorrectly predicted as positive. FN (False Negative) reflects the ratio of positive classes incorrectly predicted as negative.

Fitness function: It measures the quality of a solution calculated by the objective function (see equation (23)). It is important to note that the search process of the feature selection algorithm is guided toward high-quality solutions by this measure.

$$Fitness = \alpha * \frac{|R|}{|N|} + \beta * ERR \tag{23}$$

Selected features: It measures the ability of the proposed algorithm to select the minimum number of features that represent the entire dataset.

D. Binary-EGRO (BEGRO) for Feature Selection

In feature selection problems, the search space is binary, meaning the problem can be represented as a set of 0-1 bits. Therefore, EGRO is adjusted to work with this structure.

Transfer functions can be used to adapt optimization algorithms from continuous to binary domains without changing their core structure. The S-shaped transformation function shown in Figure 3 is represented by the Sigmoid equation (24), and its curve presents a typical smooth S shape. This function maps the continuous position values of individuals in the EGRO algorithm to the range [0, 1], and then discretizes the elements of the position vector into binary values through a random threshold mechanism.

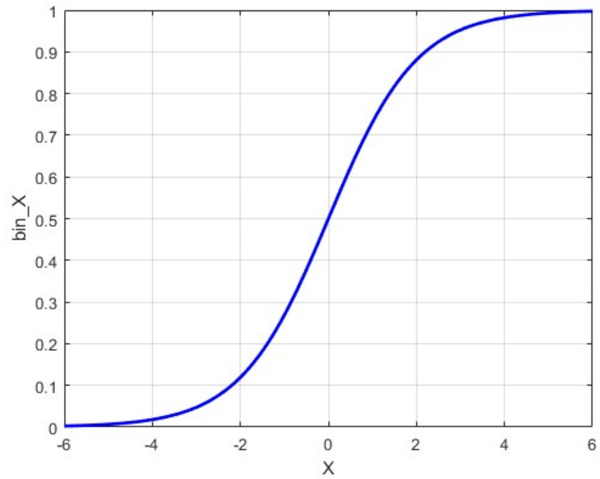

Fig. 3. S-shaped transfer function.

$$S(x) = \frac{1}{1 + \exp^{-x}} \tag{24}$$

$$Thres = 0.5 + 0.3 * \frac{\text{iter}}{\text{max}_{\text{iter}}} \tag{25}$$

By plugging the result of equations (24)(25) into equation (26), the i-th element of the gold panner's position vector is converted to 0 or 1.

$$x(t+1) = \begin{cases} 1 & , \ r < S(x(t)) \\ 0 & , \ r \geq S(x(t)) \end{cases} \tag{26}$$

where, r is a random number between 0 and 1. r plays a key role in updating the value of x_t based on the $S(x(t))$ value returned by equation (26).

As shown in Figure 4, it is divided into two parts, with arrows indicating the dynamic transition from the initial stage to the final stage. The feature selection mechanism uses the Sigmoid function to implement probability-binary conversion, with a dynamic adjustment strategy that linearly increases the threshold from 0.5 to 0.8 during the optimization process. The left side of Figure 4 represents the initial stage: lenient selection, where about 50% of features are retained to avoid the removal of important features due to random noise. The right side represents the final stage: strict selection, where only less than 20% of features are retained. This method enhances EGRO's ability to balance exploration in the feature space and discrimination power.

The effectiveness of the Binary-Enhanced Gold Rush Optimizer (BEGRO) is evaluated in this section using 16 datasets. BEGRO is compared with seven binary optimization algorithms: Binary Particle Swarm Optimization (BPSO) [36],

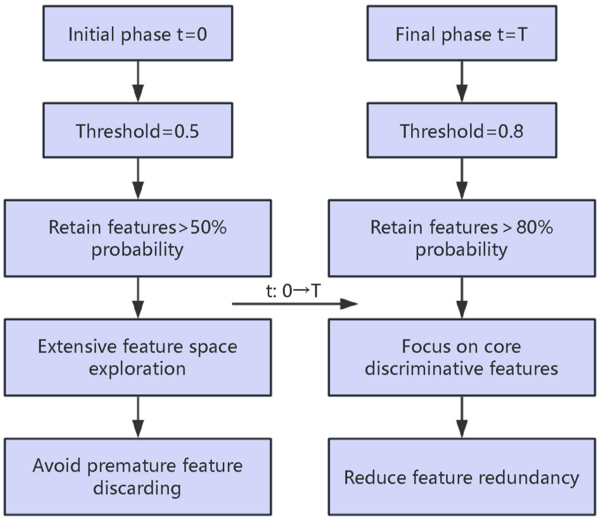

Fig. 4. Dynamic threshold feature selection mechanism.

Binary Grey Wolf Optimization (BGWO), [31] Binary Tangent Goose Optimization Algorithm (BGOA), [37] Binary Fata Morgana Optimization Algorithm (BFATA), [29] Binary Harris Hawks Optimization Algorithm (BHHO), [38] Binary Butterfly Optimization Algorithm (BBOA), [39] and Binary Gold Rush Optimization Algorithm (BGRO) [27].Each algorithm's parameters are set according to their original literature.

In this work, parameters such as the number of iterations (set to 500), population size (set to 30), and a KNN classifier utilizing Euclidean distance (K = 5) are used. To ensure a fair comparison, each method is executed 20 times. This process allows for a comprehensive evaluation of how EGRO performs relative to these other established optimization techniques on various datasets related to software fault prediction.

This section explains the effectiveness of the modifications proposed for the GRO algorithm's performance. For feature selection (FS) problems, three metrics are used for evaluation: classification accuracy, fitness value, and the number of selected features.

In Table V, we present a comparison of the classification accuracy of BEGRO and other optimization algorithms across 16 datasets. The results in Table V show that the proposed BEGRO achieves higher classification accuracy than all other optimization algorithms. BEGRO outperforms other algorithms on 81.25% of the datasets, demonstrating its effectiveness.

In Table VI, the feature selection results of each method are presented as the best fitness values from 20 runs. It is clear from Table VI that the proposed BEGRO method surpasses other methods in terms of average fitness value across 8 datasets. This indicates that the proposed optimizer achieves the best performance in 50.00% of the datasets and obtains very good results in most of the datasets.

From Table VII, BEGRO performs well in selecting the fewest features. It achieves the smallest feature count in 14 datasets. This shows that BEGRO can effectively minimize the feature set and reduce the number of features without hurting performance.

In summary, the BEGRO algorithm not only significantly reduces the number of features and lowers the model complexity in the feature selection problem of software fault prediction datasets, but also significantly improves the accuracy of the prediction model. It provides an efficient and reliable feature selection solution for software fault prediction, with important theoretical significance and practical application value.

VI. Conclusion and Future Works

To address the issues of imbalance between exploration and exploitation, slow convergence speed, susceptibility to local optima, and low convergence accuracy in the Gold Rush Optimizer (GRO), this paper proposes an improved version called the Enhanced Gold Rush Optimizer (EGRO).

First, a Sinusoidal bridging mechanism is introduced to optimize l_e, which improves the global exploration ability of GRO and accelerates the algorithm's convergence speed. Next, an adaptive Lévy flight hybrid dynamic tangent flight strategy is introduced to enhance exploration performance and population diversity. Furthermore, a partner selection strategy based on Euclidean distance is introduced, which significantly expands the search range. An adaptive Lévy flight strategy is added to further enhance exploration and improve population diversity. Finally, a Metal Detector Strategy combined with the M factor helps gold panners locate gold mines more efficiently and effectively prevents the algorithm from getting stuck in local optima.

To evaluate the performance of EGRO, experiments are conducted using CEC2022 benchmark functions. The results show that EGRO achieves excellent optimization outcomes on CEC2022 test functions, outperforming the original GRO and other comparison algorithms. Additionally, to verify the algorithm's capability in solving real-world problems, it is applied to the Software Fault Prediction problem. The experimental results show that compared to other benchmark algorithms, EGRO achieves higher classification accuracy on 81.25% of the datasets and obtains better fitness values on 93.75% of the datasets, confirming its effectiveness. Additionally, EGRO selects the minimum number of features in 87.5% of the datasets. This highlights its high optimization capability in addressing real-world practical issues.

In summary, the effectiveness of the four introduced improvement strategies is clearly demonstrated, significantly enhancing the optimization performance of the original algorithm. These strategies collaboratively improve global exploration, partner selection, population diversity, and the balance between exploration and exploitation. Moreover, EGRO proves to be highly robust, performing effectively across both unconstrained and constrained problems. Its adaptability is evident in various problem types, showcasing its versatility and broad applicability in solving complex optimization tasks. This makes EGRO a powerful tool for a wide range of real-world optimization challenges.

In future research, several areas need continuous innovation to further optimize EGRO

1) Algorithm Fusion and Optimization: Combining EGRO with other metaheuristic algorithms for fusion and col-

TABLE V Comparison of the basic and proposed BEGRO methods based on the average percent of the classification accuracy

Dataset	BPSO	BGWO	BGOA	BFATA	BHHO	BBOA	BGRO	BEGRO
Ar1	95.833	95.833	95.833	95.833	95.833	95.833	95.833	**95.833**
Ar3	100	100	100	100	100	100	100	**100**
Ar4	100	100	100	100	100	100	100	**100**
Ar5	100	100	100	100	100	100	100	**100**
Ar6	90.000	91.500	93.000	92.500	90.000	92.500	90.000	**94.500**
CM1	92.424	92.525	93.030	93.131	93.030	93.030	92.525	**93.333**
JM1	77.488	77.992	78.405	**78.488**	77.647	77.628	77.250	78.401
KC1	80.950	81.520	83.111	83.325	81.045	81.567	80.499	**83.325**
KC2	86.635	86.538	88.269	88.077	87.308	88.942	86.634	**89.808**
KC3	88.351	90.219	90.879	90.439	89.890	90.989	88.791	**91.978**
MC2	76.875	79.686	**82.187**	81.875	79.063	80.938	78.125	80.938
MW1	97.125	97.377	97.500	97.500	97.000	97.250	96.750	**97.500**
PC1	95.475	96.516	96.832	96.742	96.063	96.518	95.701	**96.968**
PC2	99.642	99.642	99.642	99.642	99.642	99.642	99.642	**99.642**
PC3	91.250	92.083	92.276	92.211	91.218	91.731	90.801	**92.404**
PC4	89.691	90.034	90.447	**90.515**	89.519	89.863	89.416	90.309
Rank	6	4	2	3	5	8	7	**1**

TABLE VI Comparison of the basic and proposed BEGRO method based on the best of the fitness value.

Dataset	BPSO	BGWO	BGOA	BFATA	BHHO	BBOA	BGRO	BEGRO
Ar1	0.045	0.044	0.043	0.044	0.043	0.042	0.044	**0.041**
Ar3	0.0036	0.0021	0.0024	0.0027	0.0010	0.0016	0.0035	**0.0003**
Ar4	0.0036	0.0023	0.0029	0.0025	0.0020	0.0022	0.0035	**0.0003**
Ar5	0.0034	0.0020	0.0023	0.0027	0.0010	0.0019	0.0032	**0.0003**
Ar6	0.1025	0.0863	0.0725	0.0774	0.1000	0.0758	0.1023	**0.0552**
CM1	0.0789	0.0768	0.0717	0.0711	0.0725	0.0707	0.0782	**0.0671**
JM1	0.2289	0.2237	0.2197	0.2184	0.2274	0.2267	0.2307	**0.2182**
KC1	0.1933	0.1872	0.1724	0.1701	0.1922	0.1867	0.1985	**0.1689**
KC2	0.1363	0.1355	0.1197	0.1223	0.1275	0.1108	0.1361	**0.1027**
KC3	0.1195	0.1008	0.0947	0.0995	0.1013	0.0911	0.1149	**0.0805**
MC2	0.2334	0.2046	**0.1809**	0.1841	0.2105	0.1909	0.2209	0.1895
MW1	0.0329	0.0296	0.0284	0.0285	0.0322	0.0309	0.0363	**0.0255**
PC1	0.0488	0.0384	0.0355	0.0360	0.4187	0.0375	0.0471	**0.0325**
PC2	0.0079	0.0069	0.0071	0.0073	0.0051	0.0064	0.0783	**0.0047**
PC3	0.0914	0.0838	0.0814	0.0824	0.0894	0.0845	0.0962	**0.0778**
PC4	0.1068	0.1036	0.0992	0.0990	0.1078	0.1029	0.1097	**0.0981**
Rank	7	6	2	3	5	4	8	**1**

TABLE VII Comparison of the basic and proposed BEGRO method based on the average number of features.

Dataset	BPSO	BGWO	BGOA	BFATA	BHHO	BBOA	BGRO	BEGRO
Ar1	11.1	7.7	7.2	7.7	3.7	2.7	10.2	**1.0**
Ar3	10.2	5.9	6.8	7.7	2.8	4.7	9.9	**1.0**
Ar4	10.2	6.6	7.0	7.1	5.8	6.2	9.9	**1.0**
Ar5	9.6	5.7	6.7	7.6	2.9	5.4	9.1	**1.0**
Ar6	10.0	6.1	9.2	8.9	2.9	4.4	9.4	**2.0**
CM1	7.8	5.5	5.4	6.1	6.9	3.3	8.4	**2.2**
JM1	12.1	11.8	11.9	10.9	12.4	10.6	11.0	**8.8**
KC1	9.5	8.5	10.5	10.2	9.2	8.5	10.9	**7.8**
KC2	8.0	4.5	7.2	8.5	3.7	**2.7**	7.6	3.6
KC3	16.1	15.2	16.9	18.5	4.9	7.3	15.2	**4.2**
MC2	17.0	13.4	17.3	17.7	12.1	8.6	16.6	**3.0**
MW1	16.3	13.1	13.3	13.7	9.0	13.0	14.9	**3.0**
PC1	8.0	7.9	8.4	7.5	5.8	6.1	9.1	**4.9**
PC2	12.2	8.7	9.2	10.1	2.3	6.9	11.9	**1.0**
PC3	17.4	19.6	18.0	19.2	**9.0**	9.6	18.5	9.4
PC4	17.1	17.9	16.6	18.5	14.7	9.2	17.7	**7.8**
Rank	8	4	7	5	3	2	6	**1**

laborative optimization fully utilizes the strengths of each algorithm. This further improves the accuracy and robustness of problem-solving.

2) Adaptability and Self-learning: By integrating machine learning and deep learning, EGRO's adaptability and self-learning abilities are enhanced, allowing it to adjust to different problems and environments. This enables EGRO to learn from data, autonomously optimize its parameters, and improve solution effectiveness.
3) Multi-objective and Constrained Optimization: As real-world problems become more complex, multi-objective and constrained optimization problems become increasingly important. Future work focuses more on EGRO's ability to solve multi-objective problems, providing more comprehensive solutions for optimization problems.
4) Expanding Application Areas: Metaheuristic algorithms are designed to solve real-world problems, and the aim is to broaden the application scope of EGRO to enhance its versatility. This includes tasks such as hyperparameter tuning in neural networks, image classification, circuit fault diagnosis, wireless sensor network optimization, and 3D point cloud reconstruction.

REFERENCES

[1] D. Albashish, A. I. Hammouri, M. Braik, J. Atwan, and S. Sahran, "Binary biogeography-based optimization based svm-rfe for feature selection," *Applied Soft Computing*, vol. 101, p. 107026, 2021.

[2] C. Miao, F. Kong, J. Zou, R. Ma, and Y. Huo, "Parallel-machine scheduling with step-deteriorating jobs to minimize the total (weighted) completion time," *Asia-Pacific Journal of Operational Research*, vol. 40, 2023.

[3] M. S. Salehi Mir, J. Rezaeian, and H. Mohamadian, "Scheduling parallel machine problem under general effects of deterioration and learning with past-sequence-dependent setup time: heuristic and meta-heuristic approaches," *Soft Computing*, vol. 24, pp. 1335–1355, 2019.

[4] J.-B. Wang, D.-Y. Lv, S.-Y. Wang, and C. Jiang, "Resource allocation scheduling with deteriorating jobs and position-dependent workloads," *Journal of Industrial Management Optimization*, vol. 19, pp. 1658–1669, 2023.

[5] L. Qi, M. Li, X. Guo, and W. Luan, "Multi-objective optimization for robotaxi dispatch with safety-carpooling mode in pandemic era," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 878–891, 2024.

[6] X. Guo, M. Zhou, S. Liu, and L. Qi, "Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption," *IEEE Transactions on Automation Science and Engineering*, vol. 18, pp. 804–816, 2021.

[7] P. Dhal and C. Azad, "A comprehensive survey on feature selection in the various fields of machine learning," *Applied Intelligence*, vol. 52, pp. 4543–4581, 2022.

[8] C. Mingbao and S. Shijie, "Two scheduling problems in group technology with deteriorating jobs," *Applied Mathematics-A Journal of Chinese Universities*, vol. 20, pp. 225–234, 2005.

[9] Y. Fu, M. Zhou, X. Guo, and L. Qi, "Artificial-molecule-based chemical reaction optimization for flow shop scheduling problem with deteriorating and learning effects," *IEEE Access*, vol. 7, pp. 53 429–53 440, 2019.

[10] F. Özyurt, "Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures," *The Journal of Supercomputing*, vol. 76, pp. 8413–8431, 2020.

[11] Y. Zhou, X. Guo, and D. Li, "A dynamic programming approach to a multi-objective disassembly line balancing problem," *Annals of Operations Research*, vol. 311, pp. 921–944, 2022.

[12] T. Xia, G. Shi, G. Si, S. Du, and L. Xi, "Energy-oriented joint optimization of machine maintenance and tool replacement in sustainable manufacturing," *Journal of Manufacturing Systems*, vol. 59, pp. 261–271, 2021.

[13] R. J. S. Raj, S. J. Shobana, I. V. Pustokhina, D. A. Pustokhin, D. Gupta, and K. Shankar, "Optimal feature selection-based medical image classification using deep learning model in internet of medical things," *IEEE Access*, vol. 8, pp. 58 006–58 017, 2020.

[14] Z. Zhang, L. Liu, J. Li, and X. Wu, "Integrating global and local feature selection for multi-label learning," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, pp. 1–37, 2023.

[15] Z. Zhang, X. Guo, M. Zhou, S. Liu, and L. Qi, "Multi-objective discrete grey wolf optimizer for solving stochastic multi-objective disassembly sequencing and line balancing problem," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 682–687.

[16] K. Jha and S. Saha, "Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique," *Applied Soft Computing*, vol. 98, p. 106823, 2021.

[17] D. Ö. Şahin, O. E. Kural, S. Akleylek, and E. Kılıç, "A novel android malware detection system: adaption of filter-based feature selection methods," *Journal of Ambient Intelligence and Humanized Computing*, vol. 46, pp. 1–15, 2023.

[18] J. Ma and X. Gao, "A filter-based feature construction and feature selection approach for classification using genetic programming," *Knowledge-Based Systems*, vol. 196, p. 105806, 2020.

[19] J. Liu, Z. Zhou, D. T. Pham, W. Xu, C. Ji, and Q. Liu, "Collaborative optimization of robotic disassembly sequence planning and robotic disassembly line balancing problem using improved discrete bees algorithm in remanufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101829, 2020.

[20] X.-f. Song, Y. Zhang, D.-w. Gong, and X.-y. Sun, "Feature selection using bare-bones particle swarm optimization with mutual information," *Pattern Recognition*, vol. 112, p. 107804, 2021.

[21] N. Maleki, Y. Zeinali, and S. T. A. Niaki, "A k-nn method for lung cancer prognosis with the use of a genetic algorithm for feature selection," *Expert Systems with Applications*, vol. 164, p. 113981, 2021.

[22] H. Zhou, J. Zhang, Y. Zhou, X. Guo, and Y. Ma, "A feature selection algorithm of decision tree based on feature weight," *Expert Systems with Applications*, vol. 164, p. 113842, 2021.

[23] G. Alp and A. F. Alkaya, "Hyperheuristic based migrating birds optimization algorithm for a fairness oriented shift scheduling problem," *Mathematical Problems in Engineering*, vol. 2021, p. 6756588, 2021.

[24] Z. Zikai, T. Qiuhua, L. Zixiang, and H. Dayong, "An efficient migrating birds optimization algorithm with idle time reduction for type-i multi-manned assembly line balancing problem," *Journal of Systems Engineering and Electronics*, vol. 32, pp. 286–296, 2021.

[25] N. Mahendran and D. R. V. PM, "A deep learning framework with an embedded-based feature selection approach for the early detection of the alzheimer's disease," *Computers in Biology and Medicine*, vol. 141, p. 105056, 2022.

[26] M. Han, "A v2g scheduling strategy based on the fruit fly optimization algorithm," in *Journal of Physics: Conference Series*, vol. 1952, 2021, p. 042063.

[27] K. Zolfi, "Gold rush optimizer: A new population-based metaheuristic algorithm," *Operations Research and Decisions*, vol. 33, pp. 113–150, 2023.

[28] X. Wang, M. Zhou, Q. Zhao, S. Liu, X. Guo, and L. Qi, "A branch and price algorithm for crane assignment and scheduling in slab yard," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1122–1133, 2020.

[29] A. Qi, D. Zhao, A. A. Heidari, L. Liu, Y. Chen, and H. Chen, "Fata: an efficient optimization method based on geophysics," *Neurocomputing*, vol. 607, p. 128289, 2024.

[30] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, pp. 341–359, 1997.

[31] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[32] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[33] H. Jia, Q. Wen, Y. Wang, and S. Mirjalili, "Catch fish optimization algorithm: a new human behavior algorithm for solving clustering problems," *Cluster Computing*, vol. 27, pp. 13 295–13 332, 2024.

[34] J. Wang, W.-c. Wang, X.-x. Hu, L. Qiu, and H.-f. Zang, "Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems," *Artificial Intelligence Review*, vol. 57, p. 98, 2024.

[35] Y. Fu, D. Liu, J. Chen, and L. He, "Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems," *Artificial Intelligence Review*, vol. 57, p. 123, 2024.

[36] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Artificial Intelligence Review*, vol. 4, pp. 1942–1948, 1995.

[37] J.-S. Pan, L.-G. Zhang, R.-B. Wang, V. Snášel, and S.-C. Chu, "Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems," *Mathematics and Computers in Simulation*, vol. 202, pp. 343–373, 2022.
[38] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Butterfly optimization algorithm: a novel approach for global optimization," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
[39] S. Arora and S. Singh, "Harris hawks optimization: Algorithm and applications," *Soft Computing*, vol. 23, pp. 715–734, 2019.

Yaxi Qing received her B.S. degree in Applied Statistics from Sichuan University of Science & Engineering, Yibin, China in 2023. She is currently pursuing the M.S. degree in Mathematics with a focus on evolutionary computation at the University of Science and Technology Liaoning, Anshan, China. Her current research interests include metaheuristic algorithms and feature selection.

Jian Zhao Received his B.S. and M.S. degrees from University of Science and Technology Liaoning, Anshan, China, in 2004 and 2007, respectively, Ph.D. degree in System Engineering from Northeastern University, Shenyang, China, in 2019. He is currently an Associate Professor of School of Science at University of Science and Technology Liaoning. His research focuses on economic dispatch, machine learning, and intelligent optimization algorithm. Till now, he has published over 20 international journal and conference papers in the above areas.