Cancer Gene Expression Data Feature Selection with ReliefF Guided Binary Pelican Optimization Algorithm

Hao-Ming Song, Jie-Sheng Wang*, Yu-Cai Wang, Yu-Wei Song, Yu-Liang Qi

Abstract—**Cancer gene expression data presents substantial challenges due to its high dimensionality, large sample sizes, and the need for multi-class classification. These complexities make efficient analysis difficult, requiring advanced techniques for meaningful data reduction and accurate classification. Feature selection (FS) is essential for addressing these challenges, as it helps minimize the dimensionality of the dataset while preserving the most crucial features for classification. In this context, this study introduces an innovative FS method, the ReliefF-guided binary Pelican Optimization Algorithm (RGBPOA), designed specifically to handle the complexities of high-dimensional cancer gene expression data. The key innovation of RGBPOA lies in integrating the ReliefF guidance strategy, which adjusts feature selection by weighing features based on their importance. This strategy allows the algorithm to effectively identify and retain the most significant features, while simultaneously eliminating redundant or irrelevant ones, thus improving classification accuracy. The approach is validated through a two-phase simulation process. In the first phase, the original Pelican Optimization Algorithm (POA) is adapted into its binary form and combined with the ReliefF strategy. Eight transfer functions are introduced to generate different algorithm variants, ensuring a broad exploration of potential configurations. The variant that demonstrates the highest performance is chosen as the most effective ReliefF-guided binary POA. In the second phase, this optimized version of RGBPOA is benchmarked against several other binary optimization algorithms to gauge its relative performance. The method is tested across 12 diverse cancer gene expression datasets, each with varying sample sizes, feature counts, and class distributions. To ensure the robustness of the results, statistical tests, including the Friedman test and Wilcoxon rank-sum test, are applied to validate the statistical significance of the findings. Ultimately, this highlights RGBPOA as a powerful and promising tool for feature selection in high-dimensional cancer gene expression data, offering significant potential for improving both the accuracy and efficiency of classification models in this domain.**

Index Terms—**Feature Selection, Pelican Optimization Algorithm, ReliefF, Cancer Gene Expression**

I. Introduction

Recent advancements in technology and measurement techniques have significantly increased the volume of data, presenting considerable challenges in fields such as cancer prediction, scientific analysis, and diagnosis. The exponential growth of data, especially in cancer research, has highlighted the need for efficient management and processing of large datasets. Cancer gene expression datasets, in particular, often involve high-dimensional features that create a vast search space, complicating both the efficiency and accuracy of data analysis [1]. These datasets typically contain numerous irrelevant or redundant features, making it difficult to extract meaningful insights. As a result, feature selection (FS) has become a critical step in cancer gene expression analysis, aimed at identifying the most relevant genes while discarding the irrelevant ones. Developing efficient methods that use minimal subsets of genes to accurately classify samples is crucial for improving the performance of these analyses [2].

Data mining plays a crucial role in extracting valuable insights from large datasets, particularly in identifying important genes within complex biological data. Feature selection (FS) techniques are essential for reducing the dimensionality of data, making it more manageable and allowing for more efficient analysis of high-dimensional datasets [3]. The main goal of FS is to reduce the number of features while maintaining or improving classification accuracy. To achieve this, FS methods often rely on binary optimization approaches, which help streamline the feature selection process by minimizing the number of features and simultaneously enhancing classifier performance [4]. In high-dimensional datasets, where the number of features can be immense, the search space for finding the optimal set of features becomes vast and increasingly difficult to navigate [5]. This challenge highlights the need for sophisticated search strategies that can enhance the efficiency and effectiveness of the FS process, enabling better identification of the most relevant features for classification tasks.

A commonly adopted approach to solve this issue involves the use of heuristic algorithms, which provide practical solutions within reasonable time and space limits [6]. Although these algorithms do not always guarantee the

Manuscript received March 23, 2025; revised May 17, 2025. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJ222410146054), and Postgraduate Education Reform Project of Liaoning Province (Grant No. LNYJG2022137).

Hao-Ming Song is a Ph. D. student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: shm@stu.ustl.edu.cn).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wjs@ustl.edu.cn).

Yu-Cai Wang is a Ph. D. student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: wyc@stu.ustl.edu.cn).

Yu-Wei Song is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: syw@stu.ustl.edu.cn).

Yu-Liang Qi is a postgraduate student at School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China (E-mail: qyl@stu.ustl.edu.cn).

discovery of the optimal solution, they adapt their search process based on individual or collective experiences, leading to feasible and effective solutions [7]. Various heuristic algorithms have been applied to FS problems, including Atom Search Optimization (ASO) [8], Salp Swarm Algorithm (SSA) [9], Differential Evolution (DE) [10], Grey Wolf Optimizer (GWO) [11], Harris Hawk Optimizer (HHO) [12], and Equalization Optimizer (EO) [13]. More recently, novel heuristic algorithms such as the AVOA algorithm [14] and the Pelican Optimization Algorithm (POA) [15] have emerged as promising alternatives for FS tasks.

Hybrid filter and wrapper FS methods have garnered significant attention due to their versatility and effectiveness. Wang et al. introduced a new framework for noninvasive blood pressure estimation using single-channel PPG signals, employing a hybrid filter-wrapper FS method to eliminate redundant and irrelevant features [16]. Similarly, Got et al. proposed a hybrid FS approach combining filter and wrapper techniques, utilizing the Whale Optimization Algorithm (WOA), a multi-objective algorithm designed to optimize both stages of feature selection. Their experimental results showed that this approach could generate smaller feature subsets without sacrificing classification accuracy [17]. Zhang et al. introduced the ReliefF-guided binary equilibrium optimization algorithm, integrating filter and wrapper methods to address FS challenges [18]. Despite the widespread use of the Relief algorithm in conjunction with swarm intelligence methods and classifiers, these approaches often suffer from limitations, such as susceptibility to local optima, which can prevent finding the optimal feature subset [19]. To overcome these challenges, Zhang et al. developed an interactive FS algorithm that combines filtering and wrapping methods, improving the overall efficiency of feature selection.

In a similar vein, Piri et al. proposed a binary multi-target filter-wrapper Chimpanzee Optimization-based FS method to predict the health status of COVID-19 patients [20], while Lu et al. presented a hybrid filter-wrapper method using information gain and Spearman correlation for feature evaluation, combined with a water wave optimization algorithm during the wrapping stage [21]. These hybrid approaches have proven effective in improving FS efficiency and classification performance.

In this paper, we propose a novel approach to feature selection (FS) that leverages a binary version of the Pelican Optimization Algorithm (POA), guided by the ReliefF strategy. The ReliefF method is incorporated into the POA to refine the selection process by prioritizing features based on their relevance to the classification task. This selective addition and removal of features aim to boost classification accuracy by focusing on the most impactful features.

To thoroughly assess the effectiveness of the proposed algorithm, eight distinct transfer functions are employed to guide the binary conversion process. These transfer functions help in optimizing how feature weights are mapped to binary values, which is crucial for balancing exploration and exploitation in the optimization process. Extensive simulation experiments were conducted to evaluate the performance of various algorithm variants, and the best-performing variant was then converted into its binary form.

The final algorithm, which integrates the ReliefF guidance strategy with several transfer functions, successfully reduces the number of selected features while enhancing both the fitness values and overall classification accuracy. This feature reduction not only makes the model more computationally efficient but also minimizes the risk of overfitting, particularly in high-dimensional datasets. The results indicate that this method is highly effective in improving feature selection, making it a promising tool for working with complex data, especially in areas like cancer gene expression analysis, where the data is typically characterized by many features and relatively few samples.

II. Pelican Optimization Algorithm (POA)

The Pelican Optimization Algorithm (POA) is an innovative computational technique inspired by the foraging behaviors of pelicans in nature. Recent research suggests that POA effectively mimics the dynamic hunting strategies employed by these birds when searching for and capturing prey. The algorithm mirrors the natural balance between exploration and exploitation that pelicans use in their hunting process, adapting these two components to enhance the algorithm's ability to navigate complex optimization landscapes. This dynamic adjustment enables POA to maintain a balance between searching broadly for new solutions and exploiting known good solutions, resulting in more efficient and accurate optimization.

In the POA framework, a group of pelicans foraging in the environment is simulated, with each pelican representing a potential solution to the optimization problem. The collective behavior of the group, driven by interactions between individual pelicans, plays a key role in guiding the group toward the optimal solution. A significant strength of POA lies in its ability to adaptively modify its search parameters based on the fitness of the solutions encountered. This self-adjustment creates a robust search mechanism that not only avoids getting trapped in local optima but also facilitates convergence toward a global optimal solution, making it particularly effective in complex optimization tasks.

When compared to traditional optimization algorithms, POA offers several advantages, including enhanced efficiency and solution quality. Its performance in a variety of optimization problems has made it a competitive and appealing choice, gaining significant attention in the research community. The POA's novel approach enriches the field of swarm intelligence and opens up new possibilities for its application in diverse areas such as engineering, data analysis, machine learning, and beyond. As the algorithm continues to evolve, it is expected to offer even more effective solutions across a broad range of domains, further advancing the capabilities of optimization techniques.

During the exploration phase of POA, pelicans identify the location of their prey and move toward that area. The movement is modeled through an equation, represented in Eq. (1), which simulates how the pelicans adjust their trajectory based on the location of the prey. This phase is critical as it ensures the algorithm explores the search space thoroughly, increasing the likelihood of finding global optima.

$$x_{i,j}^{p1} = \begin{cases} x_{i,j} + rand \cdot \left(p_j - I \cdot x_{i,j}\right) & F_p < F_j \\ x_{i,j} + rand \cdot \left(x_{i,j} - p_j\right) & else \end{cases} \quad (1)$$

where, I is a random number that can take the value of either 1 or 2.

In the POA, if the objective function value at the new position shows improvement, the pelican's position is updated accordingly. This update process is represented by Eq. (2).

$$X_{\mathrm{i}} = \begin{cases} X_i^{P1} & F_i^{P1} < F_j \\ X_{\mathrm{i}} & else \end{cases} \tag{2}$$

In the developmental phase, the prey-hunting behavior of the pelicans is modeled to enhance the local search and exploration capabilities of the POA. This process is mathematically represented by Eq. (3), where R=0.2.

$$x_{i,j}^{p2} = x_{i,j} + R \cdot \left(1 - \frac{t}{T}\right) \cdot (2 \cdot rand - 1) \cdot x_{i,j} \tag{3}$$

At this stage, valid updates are applied to either accept or reject the new positions of the pelicans. This process is represented by Eq. (4).

$$X_{\mathrm{i}} = \begin{cases} X_i^{P2} & F_i^{P2} < F_j \\ X_{\mathrm{i}} & else \end{cases} \tag{4}$$

III. ReliefF Guided Binary POA Based to Solve Feature Selection

A. Transfer Functions

A widely adopted method for converting continuous optimization problems into discrete ones is through the use of transfer functions. These functions are instrumental in transforming continuous search spaces into discrete sets, enabling more efficient and effective problem-solving. Specifically, in feature selection (FS) tasks, transfer functions play a pivotal role in evaluating and quantifying the relevance or importance of different features. By mapping the continuous feature values into a more interpretable and manageable scale, transfer functions help streamline the feature selection process, making it easier to identify the most critical features that contribute to the overall classification accuracy.

Transfer functions, especially the commonly used S-shaped and V-shaped variants, have been extensively studied in the literature for their ability to facilitate this transformation. These functions are particularly valuable in feature selection because they allow for a clear distinction between relevant and irrelevant features, which is crucial for improving the performance of machine learning algorithms. S-shaped transfer functions are known for their smooth and gradual mapping, making them suitable for problems where features exhibit a continuous but nonlinear relationship with the target variable. On the other hand, V-shaped functions provide a more abrupt transition, which can be useful when features have a more discrete or binary relevance to the classification task.

In this research, we examined eight different transfer functions, categorized into two groups: S and V-shaped. These functions were specifically selected to assess their performance in the context of feature selection (FS) tasks. By integrating these transfer functions into the FS process and running simulations, we aimed to understand how they affect both the efficiency and accuracy of the algorithm. The outcomes of these simulations offer crucial insights into the role of various transfer functions in the feature selection process, highlighting their potential to enhance the identification of relevant features and improve the overall effectiveness of FS methods. The first operation involves initializing the population and changing the input quantity to binary encoding, as shown below:

$$X_i^d = \begin{cases} 1, \ if\ rand > 0.5 \\ 0, \ else \end{cases}, i = 1,2, \ldots N \text{ , } d = 1,2, \ldots, D \tag{5}$$

The following discussion delves into the process of transforming the continuous Pelican Optimization Algorithm (POA) into its binary form by applying S-type and V-type transfer functions. The S-shaped transfer function, which is characterized by its Sigmoid curve, provides a smooth transition between low and high feature importance levels. This type of transfer function is particularly effective in feature selection (FS) tasks, as it enables the algorithm to dynamically adjust the selection of features based on their relevance, while maintaining stability. The S-type transfer function, as represented in Eq. (6), is a key component of the transfer function family, and it plays a central role in updating the concentration of particles (or potential solutions) by utilizing probability values derived from Eq. (7).

One of the primary advantages of the S-shaped transfer function is its ability to place greater emphasis on features with moderate importance, while simultaneously reducing the influence of extreme values. This characteristic makes the function highly beneficial in scenarios where the dataset includes noisy or irrelevant features, as it prevents extreme values from disproportionately affecting the feature selection process. By providing a gradual, controlled response to variations in feature relevance, the S-shaped function enhances the overall stability of the FS process and ensures a more balanced and effective selection of features. This approach is particularly useful in high-dimensional data, where many features may be irrelevant or noisy.

Furthermore, the integration of the S-shaped into POA conversion process optimizes the feature selection mechanism, enabling a more robust approach for handling complex optimization tasks. In addition to the S-shaped, the V-shaped also plays a critical role in the binary conversion of POA. Together, these transfer functions provide a versatile framework for effectively managing feature selection, enhancing both the accuracy and efficiency of the algorithm.

Table I presents the mathematical formulas for both the S and V-shaped transfer functions, while Fig. 1 illustrates their respective schematics, offering a visual representation of how these functions operate within the binary POA framework. By incorporating these transfer functions, the transformation of POA into its binary form is refined, leading to a more powerful and effective FS mechanism that is capable of handling the complexities inherent in real-world datasets.

$$T\left(X_i^d\right) = \frac{1}{1+e^{-X_i^d}} \tag{6}$$

$$X_{t+1}^d = \begin{cases} 1, \ If\ rand < \ T\left(X_t^d\right) \\ 0, \ If\ rand \geq T\left(X_t^d\right) \end{cases} \tag{7}$$

The V-shaped transfer function offers a more abrupt transition compared to its S-shaped counterpart, making it highly effective in prioritizing features that meet a specific threshold of importance while discarding those that do not meet the criteria. This sharp distinction between relevant and irrelevant features is particularly advantageous in situations where decisive feature differentiation is crucial. The V-shaped function quickly converges on the most pertinent features, streamlining the selection process by efficiently focusing on those that contribute most to the overall optimization task. Its binary-like response is especially useful when working with high-dimensional datasets, as it accelerates the dimensionality reduction process by filtering out less relevant or redundant features.

The V-shaped transfer function's sharp transition ensures that only the most significant features are selected, which is critical when dealing with large-scale datasets where irrelevant or noisy data could otherwise degrade the performance of the optimization algorithm. By employing the mathematical framework provided by Eq. (6) and Eq. (7), continuous variables are effectively mapped to binary variables, enabling a smooth conversion from a continuous search space to a discrete one. A prominent example of a V-type transfer function is the hyperbolic tangent function, which exhibits the desired binary-like behavior, making it particularly suitable for feature selection tasks.

Furthermore, the update process for the particle concentration is governed by probability values calculated from Eq. (7). This mechanism ensures that the algorithm remains focused on the most pertinent features by iteratively refining the feature subset throughout the optimization. By integrating the sharp transition behavior of the V-shaped transfer function with this dynamic updating process, the algorithm’s performance is significantly improved. This combination enhances the overall efficiency of the feature selection process, making it more capable of handling high-dimensional datasets. As a result, the approach not only helps in identifying the most important features but also ensures that the selection process is both swift and precise, making it well-suited for complex optimization challenges.

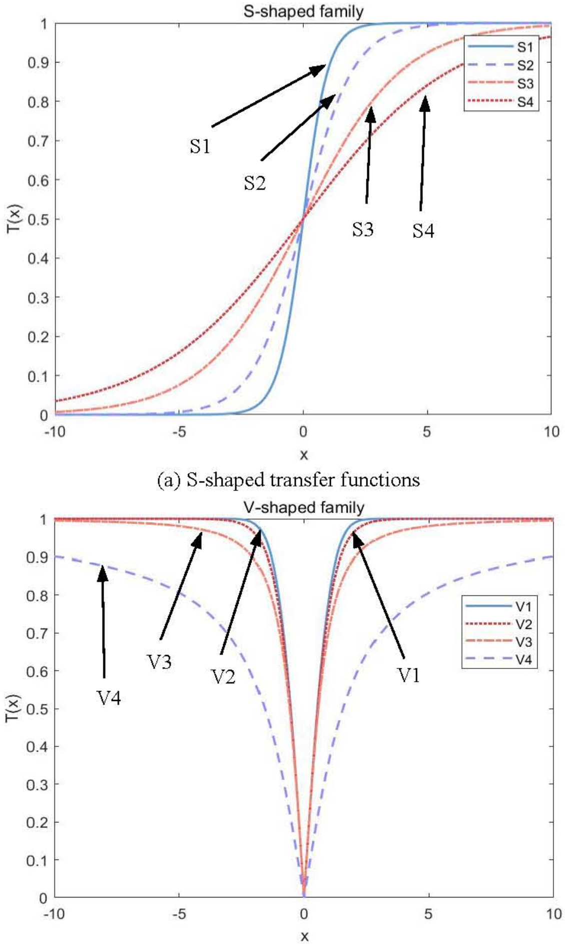
$$T\left(X_i^d\right) = \left|tanh\left(X_i^d\right)\right| \tag{8}$$

$$X_{t+1}^d = \begin{cases} -X_i^d, \ If\ rand < T\left(X_t^d\right) \\ X_i^d, \ If\ rand \geq T\left(X_t^d\right) \end{cases} \tag{9}$$

B. *ReliefF Guided Strategy*

The ReliefF algorithm is a well-established filtering feature selection (FS) method that assigns a weight to each feature, with these weights serving as indicators of feature importance. By evaluating these weights, it becomes possible to rank features according to their relevance to the classification task. However, the ReliefF algorithm has limitations, particularly due to the independence of classification algorithms, which makes it challenging to capture the combinatorial interactions between features. This limitation can restrict the algorithm’s effectiveness, especially when dealing with complex datasets in future applications. In order to address this issue, a ReliefF-guided strategy has been introduced to enhance the traditional ReliefF algorithm. This strategy aims to incorporate features with higher weights and eliminate those with lower weights, which not only improves classification accuracy but also reduces redundancy, thereby enhancing the performance of feature selection.

The ReliefF-guided approach utilizes a bootstrap strategy that involves two primary processes: feature addition and feature removal. Initially, a weight vector is constructed, which contains the weights for all features calculated through the ReliefF algorithm. This weight vector serves as the foundation for evaluating the importance of features. During the feature selection phase, a binary position vector is generated using the Particle Optimization Algorithm (POA), which reflects the current selection of features. The weights of the unselected features in the binary vector are then computed according to a specified equation shown in Eq. (10). To minimize errors during this update process, the feature corresponding to the highest median weight in the weight vector is assigned a high probability of inclusion, while the feature with the second highest median weight is assigned a slightly lower probability.



(a) S-shaped transfer functions

(b) V-shaped transfer functions

Fig. 1 S-shaped and V-shaped transfer functions.

TABLE I. S-SHAPED AND V-SHAPED TRANSFER FUNCTIONS

S-shaped and V-shaped Transfer Functions	
Function ID	Equations
S1	$T(x) = \frac{1}{1+e^{-2x}}$
S2	$T(x) = \frac{1}{1+e^{-x}}$
S3	$T(x) = \frac{1}{1+e^{\left(-\frac{x}{2}\right)}}$
S4	$T(x) = \frac{1}{1+e^{\left(-\frac{x}{3}\right)}}$
V1	$T(x) = \left\|\mathrm{erf}\left(\frac{\sqrt{\pi}}{2}x\right)\right\|$
V2	$T(x) = \|tanh(x)\|$
V3	$T(x) = \|\frac{x}{\sqrt{1+x^2}}\|$
V4	$T(x) = \left\|\frac{\pi}{2}\arctan\left(\frac{2}{\pi}x\right)\right\|$

The newly generated particle, which includes the selected feature, is then evaluated against the original particle. If the new particle results in a lower fitness value, the feature is retained, and the guided feature addition process continues as described in Eq. (11). This iterative approach ensures that features with greater relevance are gradually incorporated into the solution, leading to enhanced classification accuracy and overall performance.

Conversely, the feature removal process focuses on reducing the dimensionality of the feature set while striving to maintain or even enhance classification accuracy. This process mirrors the feature addition step, with the weights of unselected features in the binary vector being recalculated according to Eq. (12). To prevent issues related to feature dependency and ensure smooth updates, the feature with the lowest median weight is assigned a high probability of being excluded, while the feature with the second lowest weight is given a slightly lower exclusion probability. After generating the updated particle, it is compared to the original particle. If the new particle's fitness value is equal to or better than that of the original, the removed features are retained in the subset. However, if the fitness value of the updated particle is worse, the decision to exclude the features is confirmed, and those features are permanently removed from the subset. This feature removal procedure is formalized in Eq. (13), ensuring that the feature selection process eliminates irrelevant or redundant features and refines the feature set.

The integration of the ReliefF-guided strategy with the feature addition and removal processes creates a more adaptable and efficient feature selection approach. By carefully balancing the inclusion and exclusion of features based on their relevance, this method not only improves classification performance but also simplifies the feature set, making it particularly advantageous when dealing with high-dimensional data. This approach enhances the robustness of the feature selection process, ensuring that only the most important features remain for further analysis, ultimately leading to a more effective and accurate model.

$$V_{uw} = \bar{X}_i \otimes V_{weight} \tag{10}$$

$$X_i = \begin{cases} X_i(index_{(max1(V_{uw}))}) = 1, \ rand \geq \frac{1}{3} \\ X_i(index_{(max2(V_{uw}))}) = 1, \ rand < \frac{1}{3} \end{cases} \tag{11}$$

$$V_{sw} = X_i \otimes V_{weight} \tag{12}$$

$$X_i = \begin{cases} X_i(index_{(min1(V_{sw}))}) = 0, \ rand \geq \frac{1}{3} \\ X_i(index_{(min2(V_{sw}))}) = 0, \ rand < \frac{1}{3} \end{cases} \tag{13}$$

C. K-Nearest Neighbor Classifier (KNN)

K-Nearest Neighbors (KNN) is a fundamental and widely-used supervised learning algorithm known for its simplicity and versatility in solving both classification and regression tasks. The core idea of KNN is based on the assumption that similar data points are likely to have the same class or value. Essentially, it predicts the outcome of a new data point by examining the outcomes of its closest neighbors, making it particularly useful when there is a notion of locality in the data.

To apply KNN, the process begins with a labeled dataset, where each data point consists of feature values and a corresponding class label or continuous value. The first step is to choose the value of *K*, which determines how many neighboring data points will be considered when making predictions. This parameter is crucial as it affects the sensitivity of the model; a smaller *K* might make the model too sensitive to noise, while a larger *K* could oversmooth the predictions, leading to a loss of detail in the results.

After selecting *K*, the algorithm computes the distance between the new data point and every other point in the training set. The most commonly used distance metrics include Euclidean, Manhattan, and Minkowski distances, which differ in how they calculate the "closeness" between points. Once the distances are computed, the algorithm identifies the *K* closest neighbors, which are the training data points with the smallest distances to the new point.

For classification tasks, the algorithm uses a majority voting system, where the class label that appears most frequently among the *K* neighbors is assigned to the new data point. In regression tasks, the prediction is typically made by averaging the values of the *K* closest neighbors, or using a weighted average where closer neighbors have more influence on the final prediction.

One of the key strengths of KNN is its ease of understanding and implementation, making it a popular choice for simple machine learning tasks. However, KNN can become computationally expensive, particularly when dealing with large datasets. This is because the algorithm must compute distances to all points in the dataset for every new prediction. To address this issue, techniques like dimensionality reduction, data indexing methods such as KD-Trees, or approximations like Ball Trees are often used to speed up the distance calculations and improve efficiency.

Despite its potential drawbacks, particularly in terms of computational cost, KNN remains one of the most intuitive and widely used algorithms in machine learning, particularly when the data is not too large or complex. Its ability to adapt to new data without needing to retrain a model makes it particularly valuable in many real-world applications.

D. Fitness Function

In this experiment, the primary goal is to focus on minimizing classification errors, rather than merely maximizing accuracy. While reducing the number of features is important for improving computational efficiency and reducing the risk of over-fitting, the main objective is to ensure that the selected features contribute meaningfully to the model's predictive power. It is understood that a smaller feature set could potentially increase classification errors if critical features are removed, either due to their relevance or importance. Therefore, the feature selection process aims to strike a balance between dimensionality reduction and maintaining or improving classification performance, with a strong emphasis on reducing classification errors.

This approach stresses the need to not only assess features based on their individual relevance but also to consider their collective impact on the overall classification error. By evaluating the contribution of each feature, the objective is to improve the model's robustness and ability to generalize to new, unseen data. This is especially crucial in high-dimensional datasets where irrelevant, redundant, or noisy features can severely degrade performance. Consequently, the feature selection process seeks to retain only the most informative features, boosting the model's capacity to generalize and minimizing over-fitting risks.

When working with high-dimensional data, the challenge lies in distinguishing between features that genuinely aid classification and those that introduce noise. By optimizing for classification error instead of solely reducing feature count, the approach ensures that the model becomes both computationally efficient and highly accurate. This process not only enhances the model’s overall predictive power but also enables it to handle the complexity of large datasets more effectively.

IV. Experimental Results and Analysis

In this section, we detail the experimental setup and thoroughly analyze the results of our proposed feature selection algorithm. To evaluate its performance, we conducted two sets of control experiments. The first set compares several binary variants of the enhanced Pelican Optimization Algorithm (POA) using cancer gene expression datasets to identify the most effective version. After determining the best variant, it is then compared to six established optimization algorithms to assess its relative performance.

Sub-section *A* provides an extensive description of the cancer gene expression datasets, including essential attributes such as the sample size, number of features, and class distribution. The specific configuration of the experiments, including the algorithm parameters, are outlined in Sub-section *B*. Sub-section *C* describes the performance metrics used to evaluate the effectiveness of our model, ensuring a comprehensive and rigorous assessment. In Sub-section *D*, the results are analyzed in depth, comparing them to baseline feature selection methods. This comparison allows us to highlight the strengths and weaknesses of each approach, providing a clearer understanding of the advantages of the proposed algorithm over traditional methods. By evaluating the performance relative to other established algorithms, we can assess the practical applicability of the proposed method in gene expression data analysis. Additionally, the refined feature selection framework based on the enhanced POA algorithm is illustrated in Fig. 2. This figure offers a clear, visual representation of the methodology, demonstrating how the approach is implemented in practice and how the various components interact to achieve efficient feature selection.

A. Cancer Gene Expression Datasets

To assess the performance and effectiveness of the proposed feature selection (FS) algorithm applied to cancer gene expression data, we conducted an in-depth analysis using twelve diverse, high-dimensional datasets [23]. These datasets were carefully chosen to represent a wide range of scenarios, ensuring a thorough evaluation of the algorithm’s capabilities. Table II offers a detailed summary of each dataset, highlighting key characteristics such as sample size, feature count, and class distribution. The sample sizes in these datasets range from 37 to 248, while the number of features varies between 1203 and 4553. In addition, the datasets contain between 12 distinct classes, further emphasizing the complexity and variety of the data being analyzed. This diversity in dataset attributes allows for a robust and comprehensive evaluation of the proposed FS algorithm, ensuring that it is tested under a range of conditions typical in cancer gene expression data.

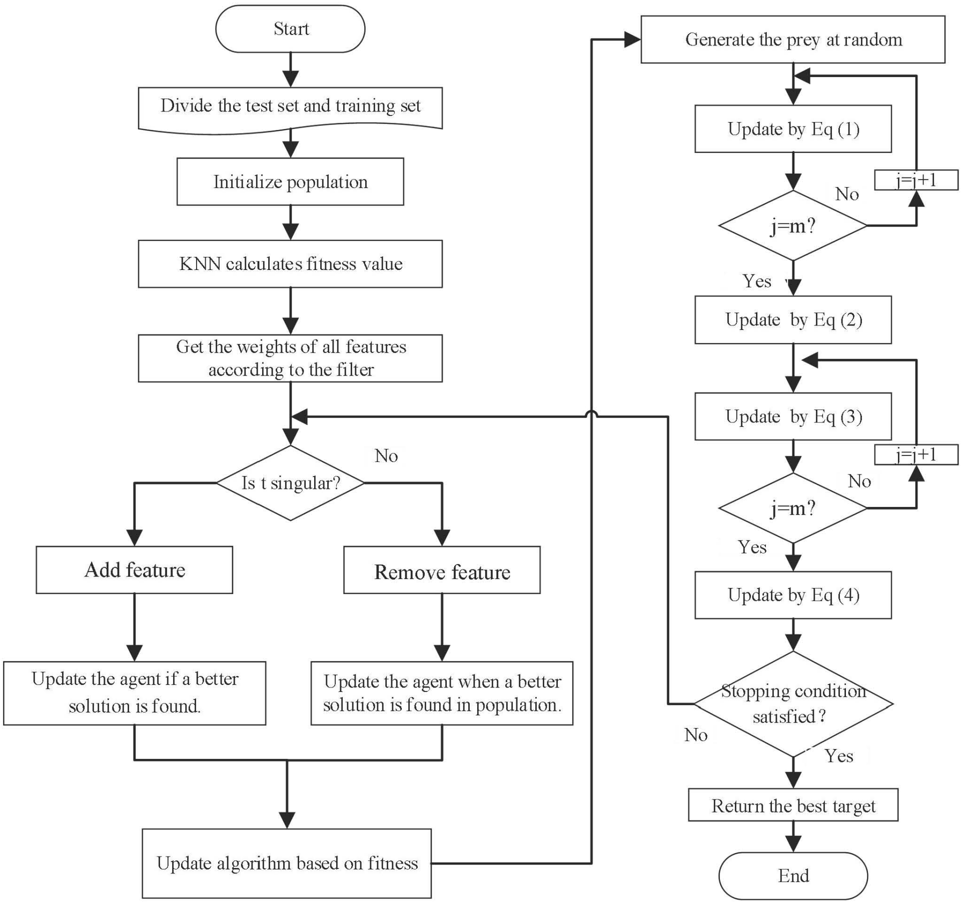

Start
Divide the test set and training set
Initialize population
KNN calculates fitness value
Get the weights of all features according to the filter
Is t singular?
No
Add feature
Remove feature
Update the agent if a better solution is found.
Update the agent when a better solution is found in population.
Update algorithm based on fitness
Generate the prey at random
Update by Eq (1)
j=j+1
No
j=m?
Yes
Update by Eq (2)
Update by Eq (3)
j=j+1
No
j=m?
Yes
Update by Eq (4)
Stopping condition satisfied?
No
Yes
Return the best target
End

Fig. 2 Flowchart of enhanced POA for feature selection.

B. Experimental Parameter Settings

To evaluate the performance of the proposed method, statistical metrics are calculated based on 30 independent experimental runs. The average of these results is used to represent the final performance of the algorithm. In this experimental setup, the population size is consistently set at 30, with a maximum iteration count of 100 to control computational resources and convergence time.

The search space is defined by the total number of features, ensuring that the algorithm explores a high-dimensional space in line with the dataset's complexity. Table III provides a comprehensive summary of the key parameters used in the experiment, detailing their specific values and explaining how these choices contribute to the overall configuration of the experiment. This table serves as a valuable reference for understanding the experimental framework and allows for replication or further adjustments in similar studies.

C. Feature Selection Performance Evaluation Criteria

Metrics are essential tools for evaluating and understanding the performance of feature selection (FS) algorithms. They help quantify how well the selected features contribute to the model ability to classify or predict. These metrics are calculated to provide a holistic view of the feature selection process. Specifically, the average classification accuracy, which is calculated using Eq. (14), indicates how accurately the selected features classify the data. The mean number of selected features, as calculated in Eq. (15), reveals how many features are chosen by the algorithm, reflecting its ability to simplify the model without sacrificing performance. The average fitness value shown in Eq. (16) evaluates the quality of the feature subset based on the classification accuracy and feature count, providing an overall measure of algorithm efficiency. Lastly, the standard deviation shown in Eq. (17) is used to assess the stability of the feature selection process, helping identify how consistently the algorithm selects high-quality feature subsets across different runs. Together, these metrics offer a comprehensive assessment of both the performance and reliability of the FS algorithm, allowing for a more nuanced understanding of its strengths and weaknesses.

$$Mean\_accuracy = \frac{1}{30}\sum_{i=1}^{30} Accuracy_i \tag{17}$$

$$Mean\_feature = \frac{1}{30}\sum_{i=1}^{30} feature_i \tag{18}$$

$$Mean\_fitness = \frac{1}{30}\sum_{i=1}^{30} fitness_i \tag{19}$$

where, $Accuracy$ is the classification accuracy.

$$Std\_fitness = \sqrt{\frac{1}{30}\sum (fitness_i - Mean\_fitness\,)^2} \tag{20}$$

D. Comparison and Discussion of Simulation Results

To comprehensively evaluate the performance of the proposed improved algorithm, the experimental process was structured in three distinct phases, each aimed at refining and optimizing the algorithm for feature selection tasks.

In the first step, the focus was on evaluating the effectiveness of the individual strategies integrated into the algorithm specifically, the African vulture satiety rate strategy and the mathematical distribution strategy. These strategies were tested independently to determine which one contributed most significantly to enhancing the algorithm’s overall performance. After running simulations and analyzing the results, the strategy demonstrating the best performance was selected for further incorporation into the algorithm. This step allowed for a focused approach to refining the optimization framework before combining strategies.

The second step involved transforming the best-performing algorithm from the first phase into its binary version to address the feature selection problem. At this stage, the ReliefF guidance strategy, which helps to prioritize relevant features based on their significance, was incorporated to further enhance the algorithm's efficiency. Additionally, a set of eight transfer functions, four S-shaped and four V-shaped, was introduced to facilitate the binary conversion process. These transfer functions allowed for a more precise mapping of continuous values to binary decisions, ensuring that the final selected features were both relevant and optimal. From this process, the improved version of the algorithm, named RGBPOA (ReliefF-guided Binary Pelican Optimization Algorithm), was chosen as the most effective for feature selection tasks.

In the final phase of the evaluation, RGBPOA was benchmarked against six other widely recognized algorithms from existing literature to assess its relative performance. To maintain fairness and ensure an unbiased comparison, all algorithms were implemented in their binary forms, providing a consistent framework for evaluating their effectiveness in feature selection tasks.

TABLE II. DETAILED INFORMATION ON 12 CANCER GENE EXPRESSION DATASETS

Number	Datasets	Subject	Genes	Classes
D(1)	Armstrong-v2	62	2093	3
D(2)	Bhattacharjee	203	1543	5
D(3)	Dyrskjor	40	1203	3
D(4)	Garber	66	4553	4
D(5)	Laiho	37	2202	2
D(6)	Lapointe-v2	110	2496	4
D(7)	Nutt-v2	50	1377	4
D(8)	Ramaswamy	190	1363	14
D(9)	Risinger	42	1771	4
D(10)	Su	174	1571	10
D(11)	Tomlins-v1	92	1288	4
D(12)	Yeoh-v2	248	2526	6

TABLE III. ALGORITHM PARAMETERS

Algorithm	Parameters	Values
ASO	α	50
	β	0.2
SSA	v_0	0
DE	CR	0.9
GWO	a	[2,0]
EO	$a1$	2
	$a2$	1
	GP	0.5

These metrics were chosen to provide a comprehensive view of each algorithm strengths, highlighting both their performance and resource usage. The purpose of this detailed comparison was to thoroughly assess the advantages and limitations of RGBPOA, validating its superior performance across various benchmark datasets. By examining these key metrics, we were able to ensure that RGBPOA's effectiveness was consistently demonstrated, offering clear evidence of its advantages over the other tested algorithms.

By conducting this multi-step experimental process, the study aimed not only to refine the RGBPOA algorithm but also to ensure that it provided significant improvements over existing optimization algorithms for feature selection. The results obtained from this approach provide compelling evidence of the effectiveness of RGBPOA, positioning it as a powerful tool for high-dimensional data analysis.

1) ReliefF Guided Novel Binary POA to Solve Feature Selection Problem

RGBPOA was introduced by incorporating a ReliefF-guided strategy to significantly improve its performance in feature selection tasks. To determine the most effective binary version of POA for feature selection, eight different variants of RGBPOA were developed, each leveraging both S-type and V-type transfer functions. These variants were rigorously tested and compared across multiple datasets to evaluate their performance.

The outcomes of these simulations, which are detailed in Tables IV-VII, offer a detailed analysis of the algorithm's effectiveness, with the top-performing results clearly highlighted in bold for easy identification. These tables provide insight into how each variant performed under various conditions, showcasing the strengths and potential areas of improvement for each configuration. To complement this, Fig. 3 displays the convergence curves of the different optimization algorithms, allowing for a visual comparison of their performance trends over time. This graphical representation makes it easier to track the progress and stability of each algorithm throughout the optimization process, providing valuable insights into their behavior and efficiency.

Table IV summarizes the mean and standard deviation of the fitness values achieved by each variant. Among these, RGBPOAS2 achieved the highest average fitness across three different datasets, closely followed by RGBPOAS4, RGBPOAV1, RGBPOAV2, and RGBPOAV3, which performed well in two datasets. RGBPOAV1, noted for its lower standard deviation in fitness values, excelled in four datasets, with RGBPOAS4 and RGBPOAV2 trailing closely, performing best in two datasets each. Analyzing both the average fitness values and their standard deviations, RGBPOAV2 emerged as the most consistent performer. Although it wasn't the top performer in every individual dataset, RGBPOAV2 demonstrated the best overall stability and yielded the most reliable comprehensive results.

In Table V, the average classification accuracy for each algorithm is presented, showing that RGBPOAV2 outperformed all other algorithms in three datasets, earning the highest overall ranking. Additionally, RGBPOAS2 and RGBPOAV2 achieved the highest classification accuracy in three datasets each. Table VII reveals that RGBPOAS2 had a clear advantage in terms of execution time, making it a suitable choice for scenarios where computational efficiency is a priority.

To visually compare the performance of the various binary POA variants, line stacked plots of the average classification accuracy rankings was plotted. This graph offers a clearer and more intuitive representation of the differences between the algorithms. Fig. 4 presents the average fitness value rankings for each algorithm after 30 independent runs. The higher the ranking of the variant, the smaller the area of the graph represented. From the simulation results, it is clear that while RGBPOAS2 performed best on many datasets, its results were more erratic in certain cases. In contrast, RGBPOAV2 demonstrated robust convergence across the majority of datasets, ranking in the top 4 of 10 datasets and in the top 3 of 7 datasets. This consistent performance suggests that RGBPOAV2 enhances both the exploration and exploitation capabilities of the algorithm, while retaining the key strengths of the original POA.

In conclusion, RGBPOAV2 emerged as the most effective binary version of the algorithm, offering a well-rounded balance between classification accuracy, feature reduction, and computational efficiency. As a result, RGBPOA with the V2 transfer function will be utilized for the control experiments in the subsequent sections of this paper. This choice ensures that the experiments are based on the most reliable and performant version of the algorithm, providing a solid foundation for further analysis and comparison.

2) RGBPOA and Other Binary Intelligent Optimization Algorithms for Cancer Gene Expression Data Feature Selection Problem

To thoroughly assess the effectiveness and benefits of the proposed RGBPOA for feature selection, an extensive comparison was conducted against several established benchmark algorithms, including BPOA, BASO, BSSA, BDE, BGWO, BHHO and BEO. This comparison spanned twelve different cancer gene expression datasets, ensuring that the results were both comprehensive and applicable across a variety of real-world scenarios.

Tables VIII through XI provide a detailed summary of the performance of the eight algorithms across these metrics. The fitness value, which combines both the classification accuracy and the number of features selected, was used as a key indicator of the overall quality of the feature subset chosen by each algorithm. This approach allows for a more balanced assessment, as it takes into account not only the effectiveness of the classification but also the efficiency of the feature selection process. By employing these metrics, the analysis offers a holistic view of how each algorithm performs in feature selection tasks, providing valuable insights into their relative strengths and weaknesses.

Table VIII reveals that RGBPOA consistently achieves the lowest average fitness value across all datasets, indicating its superior performance relative to the other algorithms. This result underscores RGBPOA’s ability to effectively identify high-quality feature subsets that contribute to improved classification accuracy. Moreover, RGBPOA demonstrates remarkable stability, as evidenced by its significantly lower standard deviation in fitness values across most datasets. This stability highlights the algorithm’ s reliability in consistently selecting optimal features, regardless of dataset variations.

In contrast, the basic POA (without the enhancements incorporated into RGBPOA) exhibits the weakest performance overall, reinforcing the positive impact of the proposed improvements. The enhanced strategy in RGBPOA leads to a more refined feature selection process, helping it outperform POA and other benchmark algorithms in both fitness and stability. These results demonstrate that the integration of ReliefF guidance and binary optimization techniques in RGBPOA provides substantial benefits over traditional methods.

In Table IX, the average classification accuracy further solidifies the superiority of RGBPOA. It outperforms all other swarm intelligence optimization algorithms across each dataset, presenting compelling evidence of its effectiveness in accurately classifying cancer gene expression data. Additionally, in Table X, RGBPOA selects the fewest features across all datasets, showcasing its exceptional ability to reduce dimension without compromising classification performance. This makes RGBPOA an attractive option for feature selection in high-dimensional data, where reducing the number of features can significantly reduce computation time and increase model interpretability.

Table XI provides an overview of the average computational time required by RGBPOA in comparison to the other algorithms. While the incorporation of advanced strategies such as the African condor satiety rate, mathematical distribution strategy, and ReliefF guidance strategy contributes to the enhanced performance of RGBPOA, it also results in a longer execution time compared to the other algorithms. This trade-off between performance and computational cost is typical in optimization algorithms that employ more complex strategies, but the results show that the increased execution time is justified by the superior feature selection outcomes.

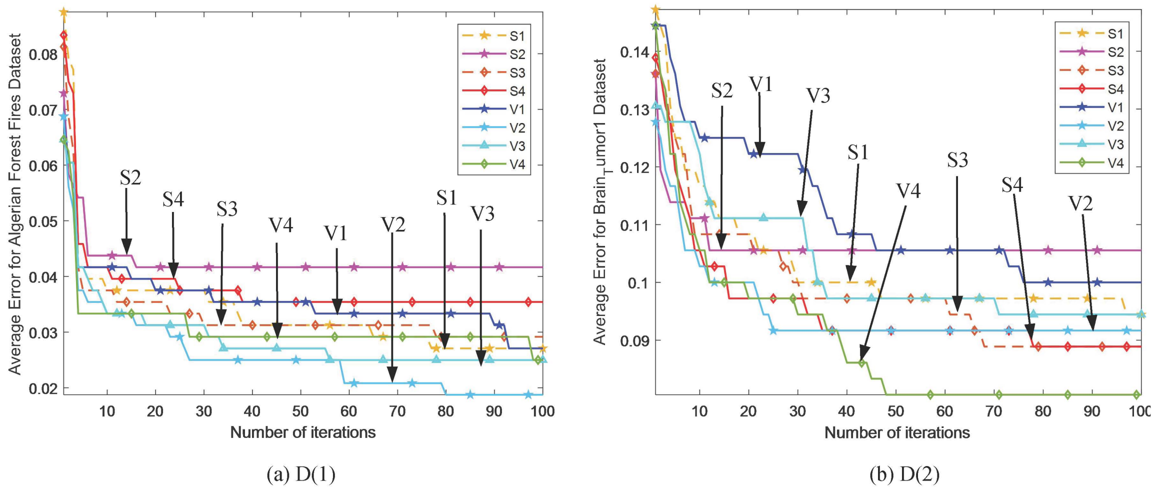
Fig. 5-6 further illustrate the performance differences between RGBPOA and the other algorithms. Fig. 5 presents the average fitness convergence curves, with the number of iterations on the horizontal axis and the average fitness values derived from thirty independent runs on the vertical axis. These results clearly demonstrate that RGBPOA consistently achieves significantly lower average fitness values across all datasets, which highlights its superior convergence rate and its ability to avoid local optima more effectively than the other algorithms.

Fig. 6 shows the in-line stacked plot of the average classification accuracy ranking, where the smaller shaded area in the corresponding graph for each algorithm represents its higher ranking. Among them, RGBPOA is the smallest in area, indicating that it performs very well on all datasets. Collectively, the results from these experiments provide compelling evidence that RGBPOA outperforms other algorithms in several critical aspects, including average fitness, classification accuracy, and the number of selected features. These outcomes strongly support the effectiveness of RGBPOA in tackling the inherent challenges of feature selection, particularly in complex, high-dimensional datasets such as those encountered in cancer gene expression analysis. The findings demonstrate that RGBPOA not only surpasses traditional swarm intelligence-based algorithms but also exhibits exceptional performance when balancing accuracy with computational efficiency. This unique combination makes it particularly well-suited for feature selection tasks, where both the quality of the feature subset and the processing time are crucial. Given its ability to manage these competing demands effectively, RGBPOA emerges as a highly promising solution for real-world data analysis challenges, offering both high classification performance and practical applicability across diverse and intricate datasets.

3) Wilcoxon Test Results Analysis

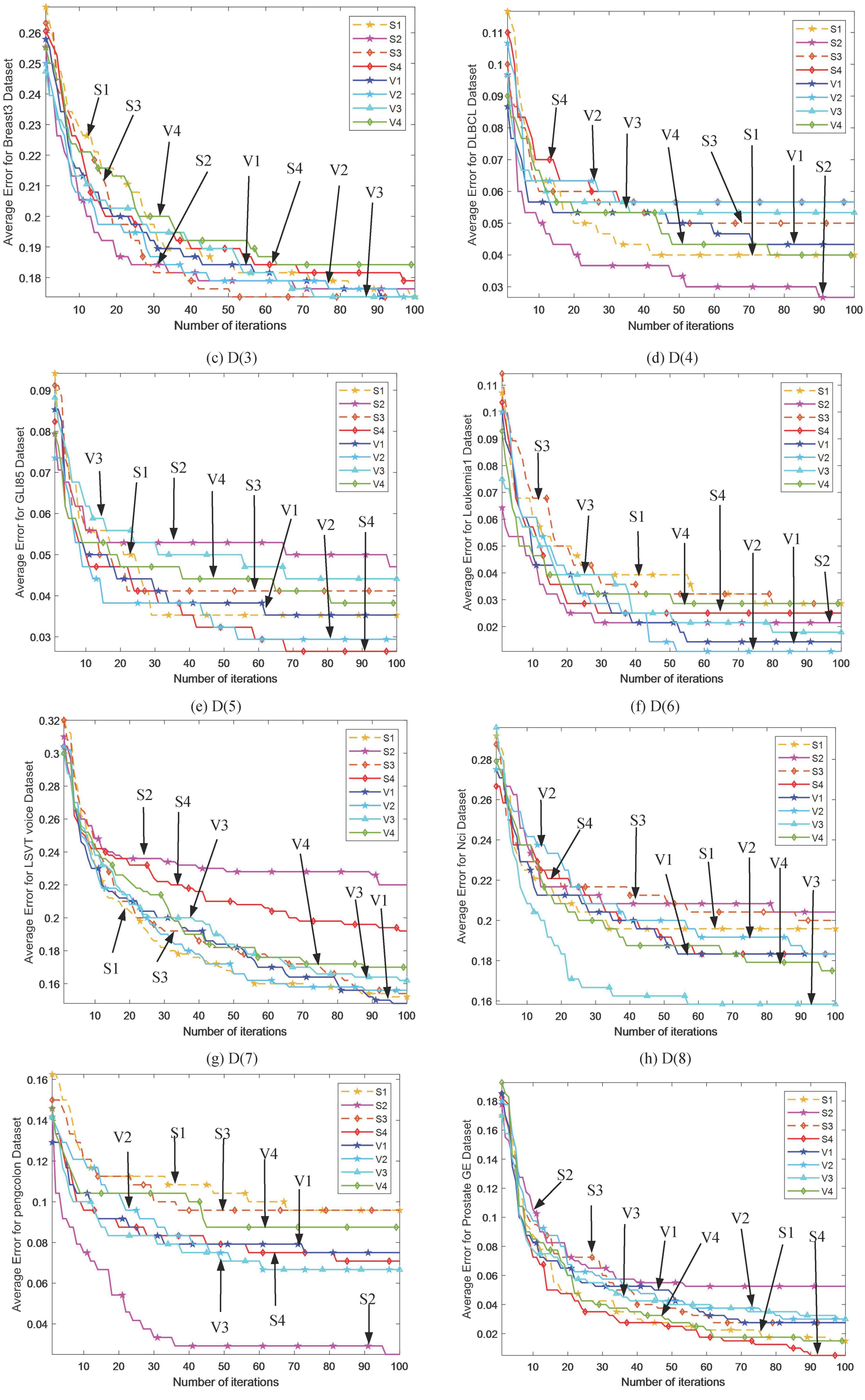
Table XII assesses the effectiveness of the proposed method through a rigorous p-value analysis, utilizing the Wilcoxon rank sum test to compare the performance of RGBPOA with that of seven alternative binary intelligent optimization algorithms. This statistical test is specifically designed to determine whether the differences in results between RGBPOA and the other algorithms are statistically significant.

The results presented in Table XII show that the p-values for all comparisons between RGBPOA and the alternative algorithms fall well below the 0.05 threshold. This indicates that the performance differences observed are not due to random variation but are statistically significant. This robust statistical evidence further validates the efficacy of RGBPOA, demonstrating that it outperforms the other binary optimization algorithms in a consistent and meaningful way across the datasets tested. This statistical confirmation strengthens the case for adopting RGBPOA in real-world feature selection tasks, providing confidence in its ability to deliver superior results in complex data analysis problems.



(a) D(1)

(b) D(2)



(c) D(3)

(d) D(4)

(e) D(5)

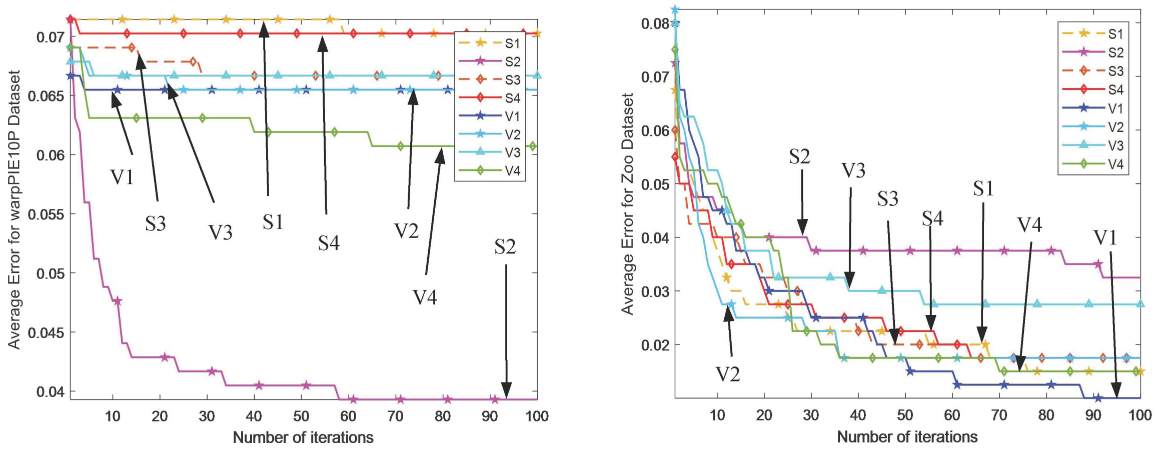
(f) D(6)

(g) D(7)

(h) D(8)

(i) D(9)

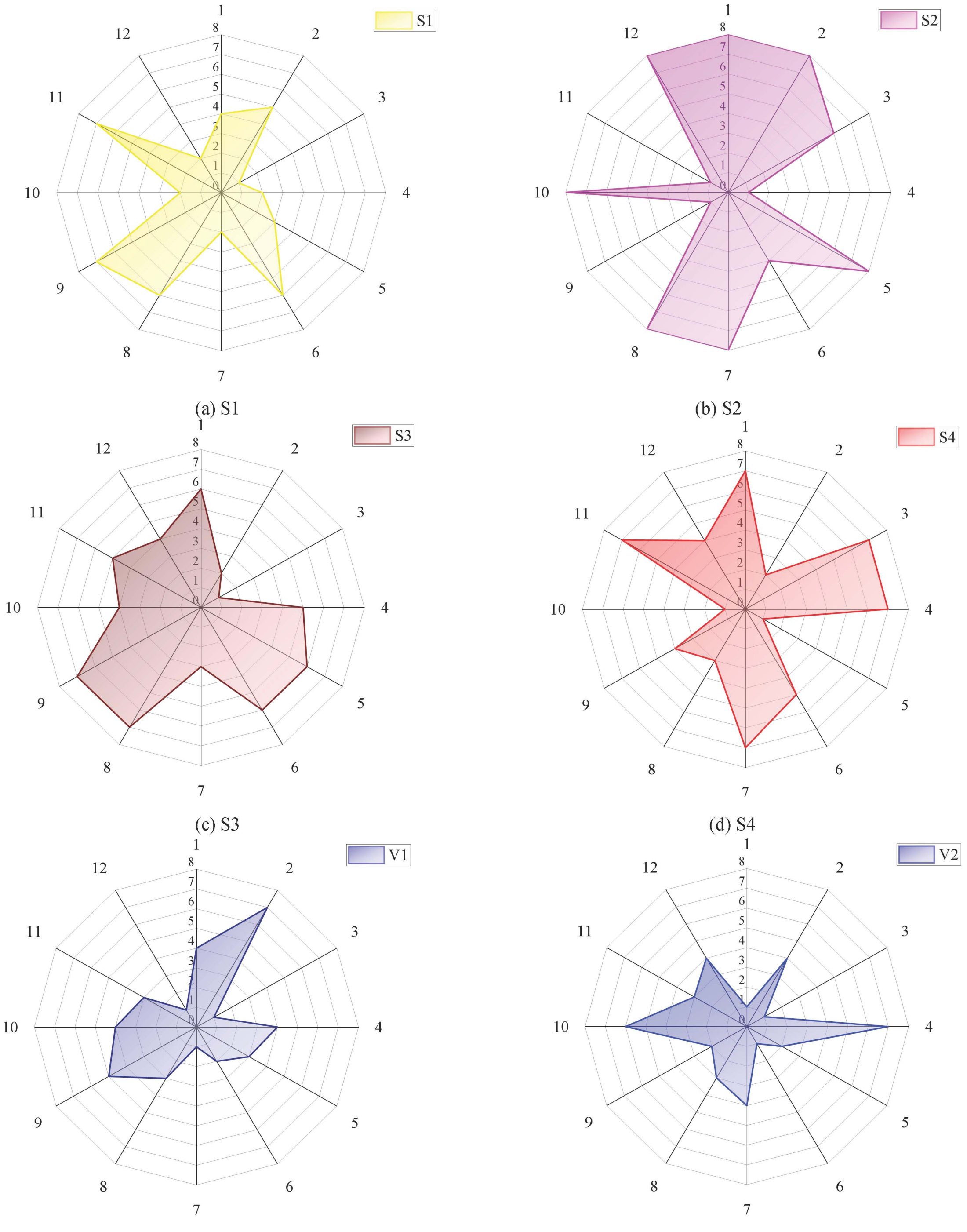
(j) D(10)



(k) D(11)

(l) D(12)

Fig. 3 Convergence curves of binary POA variants on twelve datasets.



(a) S1

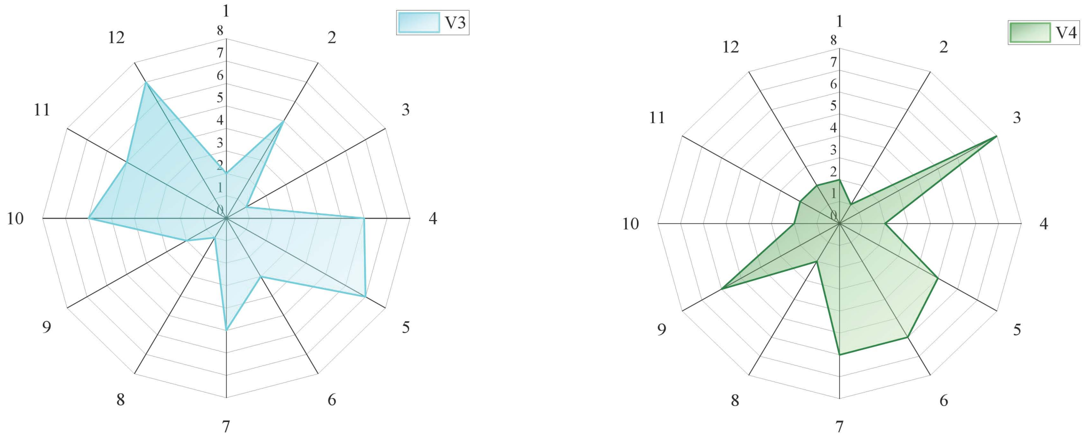
(b) S2

(c) S3

(d) S4

(e) V1

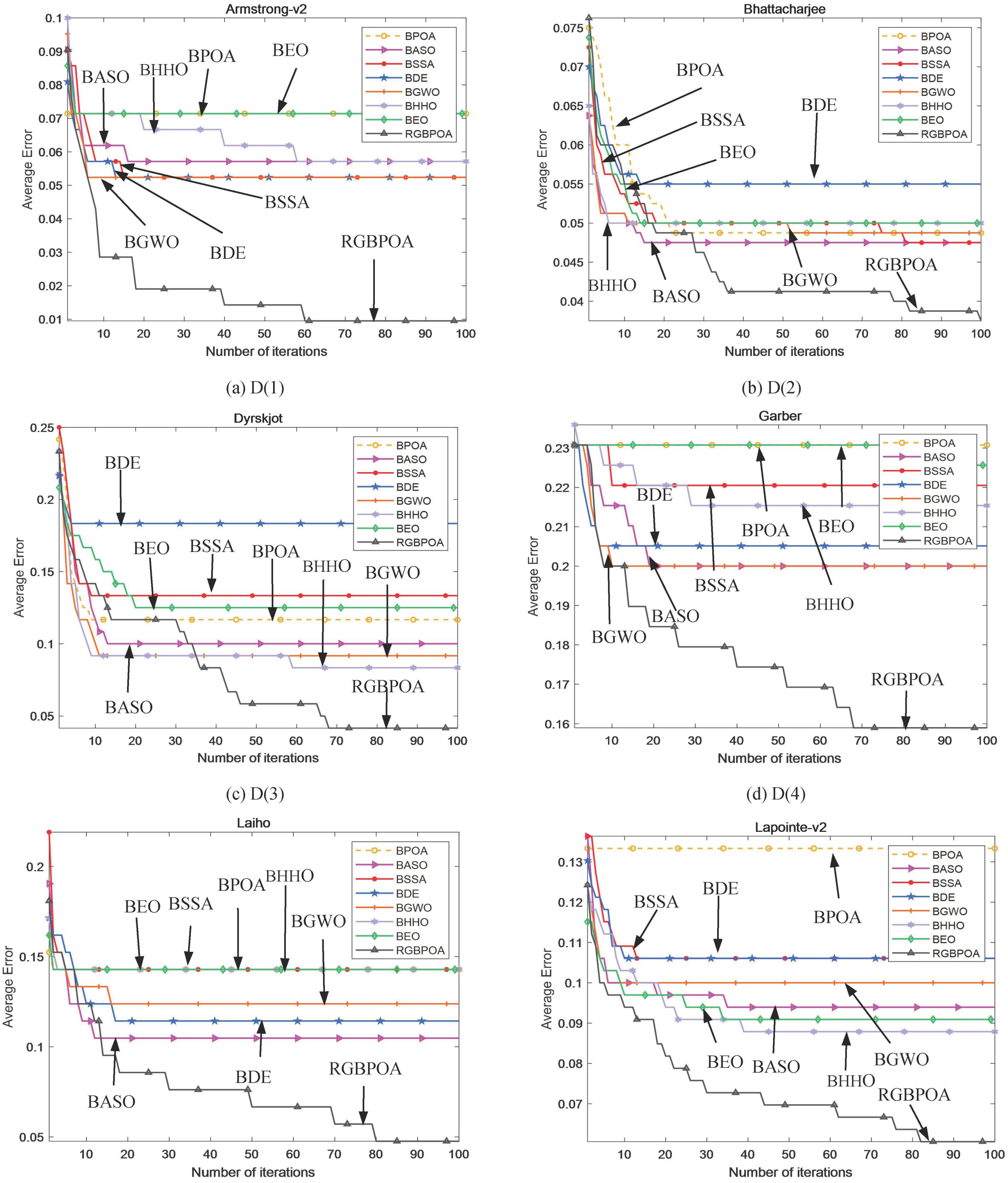
(f) V2



(g) V3

(h) V4

Fig. 4 In-line stacked plots of binary POA variants on twelve datasets.



(a) D(1)

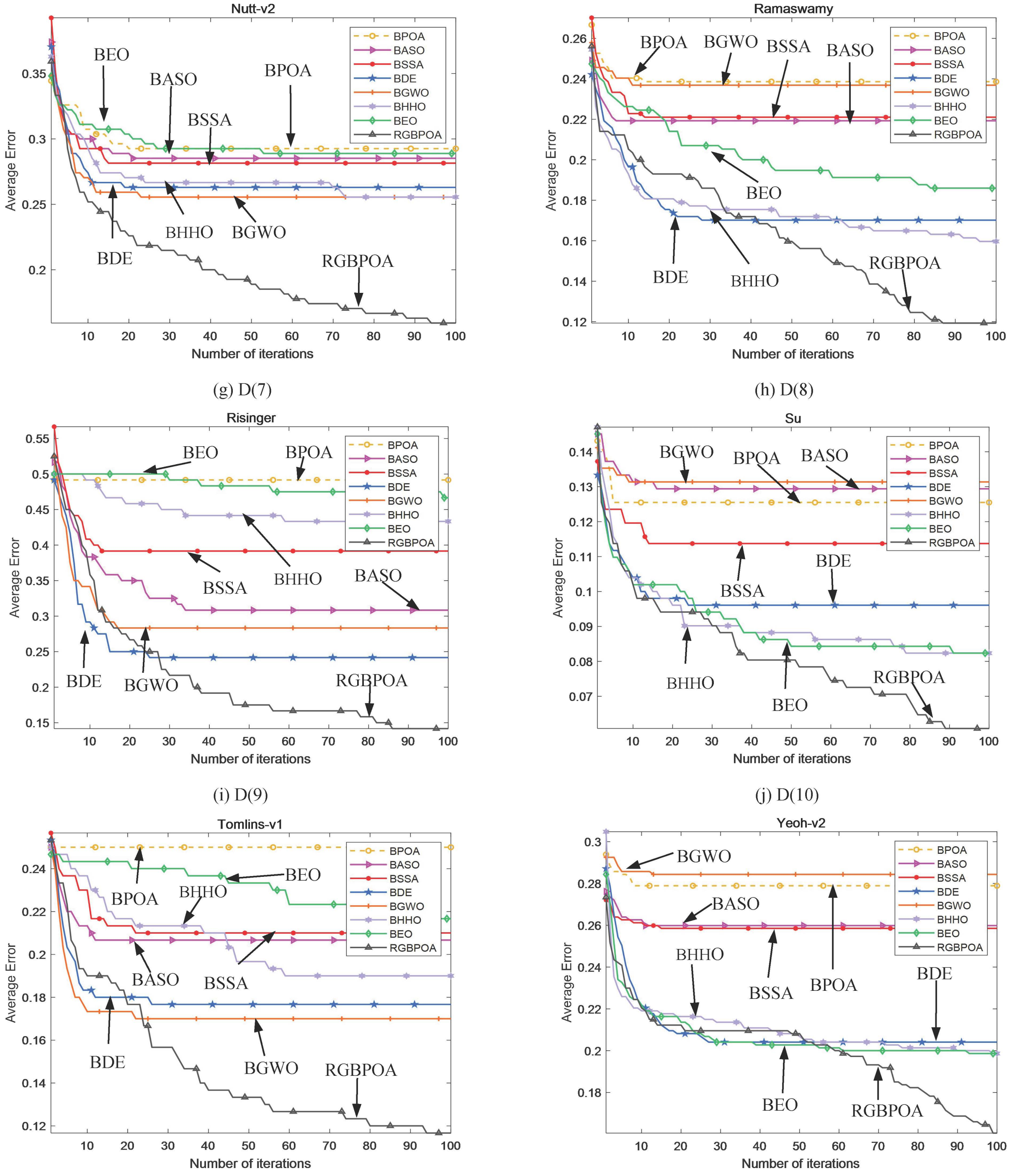
(b) D(2)

(c) D(3)

(d) D(4)

(e) D(5)

(f) D(6)



(g) D(7)

(h) D(8)

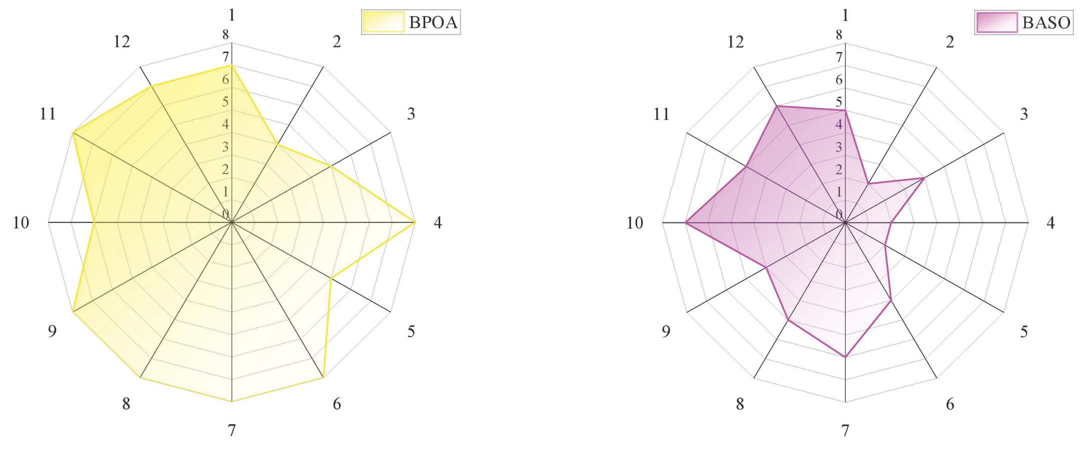
(i) D(9)

(j) D(10)

(k) D(11)

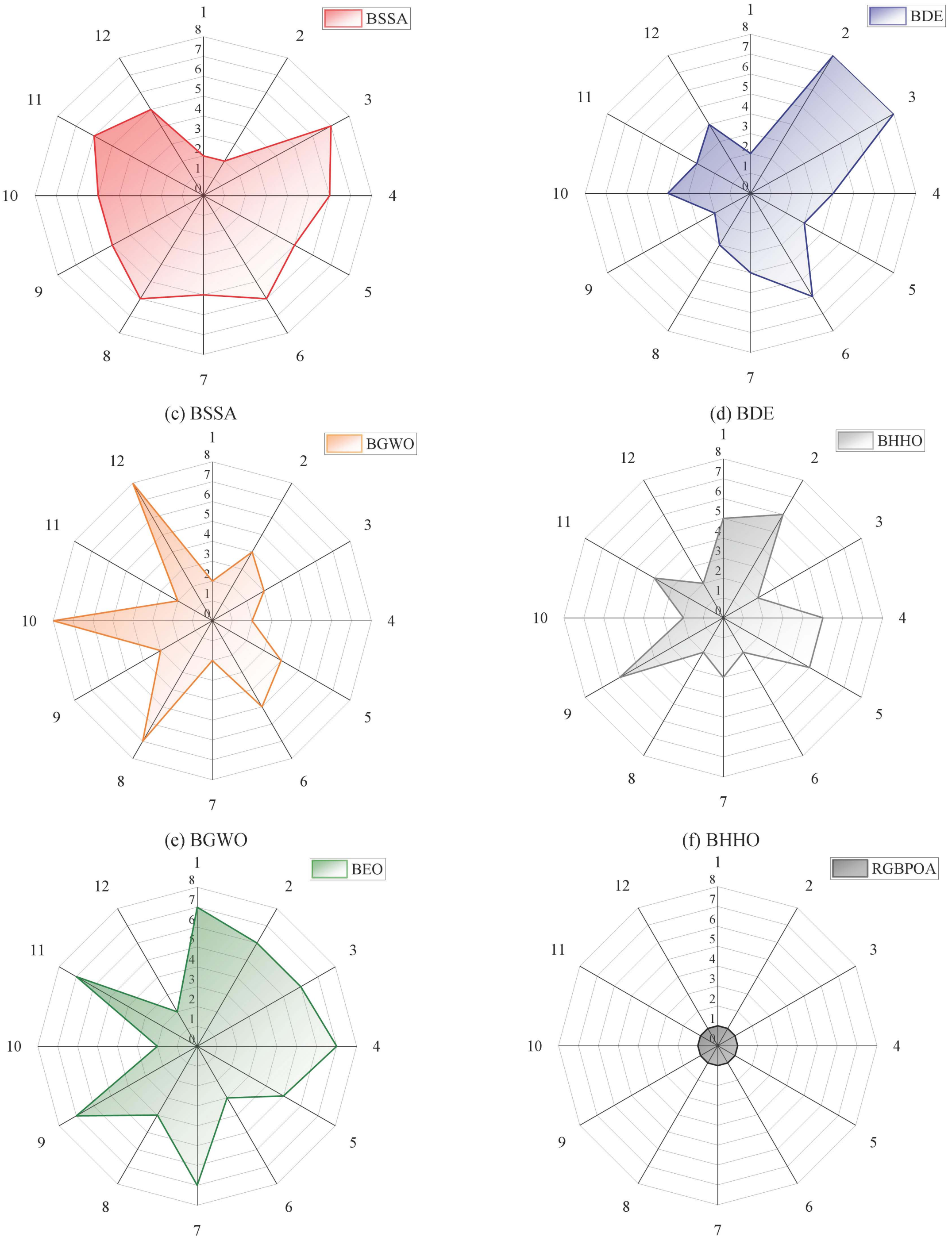
(l) D(12)

Fig. 5 Convergence curves of RGBPOA and other algorithms on cancer gene expression datasets.



(a) BPOA

(b) BASO



(c) BSSA

(d) BDE

(e) BGWO

(f) BHHO

(g) BEO

(h) RGBPOA

Fig. 6 In-line stacked plots of RGBPOA and other algorithms on cancer gene expression datasets.

TABLE IV. AVERAGE FITNESS AND STANDARD DEVIATION OF ACCURACY FOR DIFFERENT RGBPOA

Dataset	Measure	S1	S2	S3	S4	V1	V2	V3	V4
D(1)	AVG	0.0324	0.0454	0.0337	0.0407	0.0305	**0.0225**	0.0282	0.0284
	STD	0.0195	**0.0012**	0.0169	0.0105	0.0202	0.0204	0.0190	0.0207
D(2)	AVG	0.0997	0.1095	0.0936	0.0934	0.1034	0.0951	0.0976	**0.0836**
	STD	0.0314	0.0246	0.0276	0.0329	**0.0226**	0.0269	0.0259	0.0281
D(3)	AVG	0.1781	0.1796	0.1776	0.1826	0.1764	0.1763	**0.1761**	0.1862
	STD	0.0298	0.0349	0.0298	0.0311	**0.0245**	0.0298	0.0298	0.0359
D(4)	AVG	0.0458	**0.0314**	0.0551	0.0615	0.0473	0.0605	0.0569	0.0435
	STD	0.0332	0.0332	0.0293	0.0323	**0.0242**	**0.0242**	0.0271	0.0332
D(5)	AVG	0.0412	0.0516	0.0464	**0.0316**	0.0394	0.0335	0.0478	0.0417
	STD	0.0293	0.0305	0.0274	0.0297	0.0293	0.0299	**0.0259**	0.0285
D(6)	AVG	0.0345	0.0262	0.0339	0.0302	0.0186	**0.0149**	0.0218	0.0322

	STD	0.0355	0.0333	0.0356	0.0346	0.0291	**0.0259**	0.0314	0.0355
D(7)	AVG	0.1566	0.2227	0.1581	0.1955	**0.1510**	0.1587	0.1646	0.1721
	STD	0.0473	0.0472	**0.0392**	0.0524	0.0546	0.0602	0.0454	0.0559
D(8)	AVG	0.2001	0.2071	0.2036	0.1869	0.1859	0.1858	**0.1609**	0.1771
	STD	0.0484	0.0499	**0.0415**	0.0432	0.0574	0.0508	0.0529	0.0592
D(9)	AVG	0.1011	**0.0297**	0.1005	0.0755	0.0787	0.0703	0.0701	0.0905
	STD	0.0671	**0.0388**	0.0670	0.0484	0.0650	0.0633	0.0633	0.0565
D(10)	AVG	0.0211	0.0570	0.0329	**0.0104**	0.0316	0.0340	0.0338	0.0187
	STD	0.0233	0.0195	0.0340	**0.0152**	0.0299	0.0373	0.0296	0.0233
D(11)	AVG	0.0757	**0.0439**	0.0716	0.0750	0.0692	0.0692	0.0701	0.0640
	STD	**0.0053**	0.0116	0.0097	**0.0053**	0.0105	0.0129	0.0097	0.0142
D(12)	AVG	0.0211	0.0375	0.0229	0.0232	**0.0147**	0.0220	0.0323	0.0190
	STD	0.0232	0.0291	0.0242	0.0242	**0.0196**	0.0240	0.0253	0.0235
(+/=/−)		0/0/12	3/0/9	0/0/12	2/0/10	2/0/10	2/0/10	2/0/10	1/0/11
Friedman rank		5.08	5.75	5.42	5.00	3.46	**3.29**	4.25	3.75
Rank		6	8	7	5	2	1	4	3

TABLE V. AVERAGE ACCURACY OF DIFFERENT RGBPOA

Dataset	S1	S2	S3	S4	V1	V2	V3	V4
D(1)	0.9729	0.9583	0.9708	0.9646	0.9729	**0.9813**	0.9750	0.9750
D(2)	0.9056	0.8944	0.9111	0.9111	0.9000	0.9083	0.9056	**0.9194**
D(3)	**0.8263**	0.8237	**0.8263**	0.8211	**0.8263**	**0.8263**	**0.8263**	0.8158
D(4)	0.9600	**0.9733**	0.9500	0.9433	0.9567	0.9433	0.9467	0.9600
D(5)	0.9647	0.9529	0.9588	**0.9735**	0.9647	0.9706	0.9559	0.9618
D(6)	0.9714	0.9786	0.9714	0.9750	0.9857	**0.9893**	0.9821	0.9714
D(7)	0.8480	0.7800	0.8460	0.8080	**0.8520**	0.8440	0.8380	0.8300
D(8)	0.8042	0.7958	0.8000	0.8167	0.8167	0.8167	**0.8417**	0.8250
D(9)	0.9042	**0.9750**	0.9042	0.9292	0.9250	0.9333	0.9333	0.9125
D(10)	0.9850	0.9475	0.9725	**0.9950**	0.9725	0.9700	0.9700	0.9850
D(11)	0.9298	**0.9607**	0.9333	0.9298	0.9345	0.9345	0.9333	0.9393
D(12)	0.9850	0.9675	0.9825	0.9825	**0.9900**	0.9825	0.9725	0.9850
(+/=/−)	1/1/11	3/0/9	1/1/12	2/0/10	3/1/9	3/1/9	2/1/10	1/0/11
Friedman rank	4.50	3.25	3.83	4.13	5.22	**5.42**	4.46	5.08
Rank	5	8	7	6	2	1	4	3

TABLE VI. AVERAGE NUMBER OF SELECTED FEATURES OF DIFFERENT RGBPOA

Dataset	S1	S2	S3	S4	V1	V2	V3	V4
D(1)	857.30	855.40	856.30	857.30	842.80	**834.50**	842.16	845.10
D(2)	638.60	629.95	634.95	632.55	624.60	**623.30**	626.95	625.70
D(3)	730.45	724.55	732.35	689.15	702.50	**688.30**	693.75	698.20
D(4)	2091.30	1830.25	2067.25	1879.75	1864.65	**1810.70**	1815.45	1879.95
D(5)	1384.55	1115.55	1253.10	1208.80	921.80	**862.15**	989.10	965.40
D(6)	1298.05	1658.65	1992.85	1892.90	1207.45	**1069.65**	1380.25	1308.00
D(7)	889.60	850.95	876.05	868.35	731.65	**658.00**	739.15	731.25
D(8)	852.20	721.60	750.80	746.05	686.00	**638.10**	728.30	674.35
D(9)	1138.50	999.35	917.25	975.00	827.25	**673.45**	784.35	768.75
D(10)	1009.30	890.20	959.55	930.95	858.20	**809.25**	835.90	892.60
D(11)	1200.35	1015.05	1153.70	1112.90	997.80	**944.10**	1063.80	1053.85
D(12)	1410.00	1338.55	1338.85	1339.45	1328.15	**1286.65**	1307.70	1307.40
(+/=/−)	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	12/0/0	0/0/12	0/0/12
Friedman rank	7.96	5.00	6.83	6.21	2.29	**1.00**	3.79	2.92
Rank	8	5	7	6	2	1	4	2

TABLE VII. AVERAGE COMPUTATION TIME OF DIFFERENT RGBPOA

Dataset	S1	S2	S3	S4	V1	V2	V3	V4
D(1)	9.3360	**6.4536**	9.2664	9.2055	8.8320	8.7701	8.7999	8.7757
D(2)	8.0727	**4.7954**	8.1889	7.6811	8.0292	7.3175	7.6970	7.2185

D(3)	10.4284	**5.7941**	10.2642	10.2659	9.8217	9.7039	9.7258	9.6903
D(4)	11.5245	**8.7780**	11.5999	11.9038	11.1228	11.3896	11.2997	11.2984
D(5)	7.4714	**5.6507**	7.6506	7.4198	7.5308	7.7885	7.5840	7.8262
D(6)	9.2563	**6.2034**	9.1994	8.9720	8.7603	8.6911	8.7965	8.4847
D(7)	6.2933	**2.7169**	6.3029	6.2604	6.1795	6.1327	6.1263	6.0714
D(8)	8.5160	**4.1882**	8.4747	8.2783	8.0627	8.0683	7.5965	7.7193
D(9)	8.8181	**5.6467**	8.7829	8.7282	8.6828	8.6382	8.6272	8.6466
D(10)	9.7457	**7.4441**	9.4361	9.1050	8.7541	9.3101	8.3788	9.0877
D(11)	8.1700	**6.1337**	7.6391	7.9965	7.4714	7.3743	7.2387	6.6327
D(12)	11.6339	**8.3127**	11.6107	11.6092	11.4977	11.5345	11.5139	11.5588
(+/=/−)	0/0/12	12/0/0	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12
Friedman rank	7.50	**1.00**	7.08	6.00	4.08	3.67	3.67	3.00
Rank	8	1	7	6	5	3	4	2

Table VIII. Average fitness and standard deviation of accuracy of RGBPOA and other algorithms

Dataset	Measure	BPOA	BASO	BSSA	BDE	BGWO	BHHO	BEO	RGBPOA
D(1)	AVG	0.0759	0.0628	0.0590	0.0574	0.0573	0.0616	0.0759	**0.0156**
	STD	0.0291	0.0365	0.0358	0.0292	0.0345	0.0323	0.0339	**0.0248**
D(2)	AVG	0.0545	0.0511	0.0521	0.0609	0.0530	0.0550	0.0560	**0.0433**
	STD	0.0098	0.0082	0.0076	0.0095	0.0055	**0.0006**	0.0008	0.0127
D(3)	AVG	0.1211	0.1052	0.1392	0.1871	0.0962	0.0875	0.1296	**0.0463**
	STD	0.0226	0.0186	0.0206	0.0206	0.0228	**0.0011**	0.0217	0.0206
D(4)	AVG	0.2334	0.2042	0.2255	0.2087	0.2034	0.2180	0.2284	**0.1636**
	STD	0.0452	0.0743	0.0635	0.0673	0.0536	0.0631	0.0365	**0.0197**
D(5)	AVG	0.1466	0.1099	0.1486	0.1188	0.1280	0.1465	0.1465	**0.0534**
	STD	0.0311	0.0416	0.0367	0.0446	0.0495	0.0405	0.0215	**0.0185**
D(6)	AVG	0.1370	0.0992	0.1122	0.1106	0.1044	0.0922	0.0953	**0.0662**
	STD	**0.0116**	0.0267	0.0219	0.0278	0.0186	0.0121	0.0605	0.0368
D(7)	AVG	0.2960	0.2885	0.2858	0.2660	0.2585	0.2581	0.2924	**0.1639**
	STD	0.0325	0.0409	0.0529	0.0639	0.0581	0.0278	**0.0224**	0.0653
D(8)	AVG	0.2419	0.2233	0.2260	0.1757	0.2399	0.1634	0.1913	**0.1230**
	STD	0.0287	0.0235	0.0438	0.0243	0.0327	0.0179	**0.0152**	0.0239
D(9)	AVG	0.4917	0.3114	0.3949	0.2448	0.2859	0.4336	0.4670	**0.1465**
	STD	0.0320	0.1311	0.0920	0.1189	0.0734	0.0643	0.0567	**0.0435**
D(10)	AVG	0.1299	0.1343	0.1197	0.1024	0.1355	0.0879	0.0888	**0.0651**
	STD	0.0280	0.0265	0.0216	0.0207	0.0186	0.0122	**0.0120**	0.0206
D(11)	AVG	0.2525	0.2108	0.2151	0.1805	0.1737	0.1924	0.2195	**0.1217**
	STD	0.0203	0.0187	0.0174	0.0241	0.0242	0.0010	**0.0009**	0.0277
D(12)	AVG	0.2817	0.2635	0.2631	0.2103	0.2869	0.2031	0.2044	**0.1640**
	STD	0.0249	0.0291	0.0272	0.0171	0.0291	**0.0092**	0.0093	0.0263
(+/=/−)		0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	12/0/0
Friedman rank		7.13	4.50	5.58	4.33	4.25	3.63	5.58	**1.00**
Rank		8	5	6	4	3	2	6	1

Table IX. Average accuracy of RGBPOA and other algorithms

Dataset	BPOA	BASO	BSSA	BDE	BGWO	BHHO	BEO	RGBPOA
D(1)	0.9286	0.9429	0.9476	0.9476	0.9476	0.9429	0.9286	**0.9905**
D(2)	0.9513	0.9525	0.9525	0.9450	0.9513	0.9500	0.9500	**0.9625**
D(3)	0.8833	0.9000	0.8667	0.8167	0.9083	0.9167	0.8750	**0.9583**
D(4)	0.7692	0.8000	0.7795	0.7949	0.8000	0.7846	0.7744	**0.8410**
D(5)	0.8571	0.8952	0.8571	0.8857	0.8762	0.8571	0.8571	**0.9524**
D(6)	0.8667	0.9061	0.8939	0.8939	0.9000	0.9121	0.9091	**0.9394**
D(7)	0.7074	0.7148	0.7185	0.7370	0.7444	0.7444	0.7111	**0.8407**
D(8)	0.7614	0.7807	0.7789	0.8298	0.7632	0.8404	0.8140	**0.8807**
D(9)	0.5083	0.6917	0.6083	0.7583	0.7167	0.5667	0.5333	**0.8583**
D(10)	0.8745	0.8706	0.8863	0.9039	0.8686	0.9176	0.9176	**0.9392**
D(11)	0.7500	0.7933	0.7900	0.8233	0.8300	0.8100	0.7833	**0.8833**

D(12)	0.7211	0.7401	0.7415	0.7959	0.7156	0.8014	0.8014	**0.8395**
(+/=/−)	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	12/0/0
Friedman rank	1.96	4.54	3.71	4.63	4.63	5.08	3.46	**8.00**
Rank	8	4	6	3	3	2	7	1

Table X. Average number of selected features of RGBPOA and other algorithms

Dataset	BPOA	BASO	BSSA	BDE	BGWO	BHHO	BEO	RGBPOA
D(1)	1358.75	1369.40	1096.15	1097.60	1088.40	1093.45	1109.00	**833.00**
D(2)	1005.80	948.70	783.30	991.30	837.70	847.95	960.45	**624.25**
D(3)	865.33	742.60	741.60	670.47	644.00	602.20	606.87	**684.58**
D(4)	3250.73	2817.40	2628.13	2558.20	2470.87	2279.47	2765.13	**1814.47**
D(5)	1577.87	1358.80	1368.33	1237.07	1192.07	1115.47	1122.53	**834.53**
D(6)	1788.53	1549.13	1250.53	1397.80	1351.47	1296.93	1331.73	**1043.20**
D(7)	922.27	798.87	819.73	728.47	702.40	797.80	828.87	**654.07**
D(8)	973.13	844.80	974.93	986.80	740.80	733.07	982.00	**666.47**
D(9)	1097.33	878.80	953.26	991.13	960.47	812.60	882.27	**699.67**
D(10)	1128.87	968.60	1110.07	882.73	976.20	876.07	970.93	**762.73**
D(11)	1416.47	1243.47	1659.33	1290.80	1262.93	1433.93	1413.67	**951.13**
D(12)	1713.53	1564.07	1806.87	2084.67	1666.47	1624.00	1960.40	**1277.67**
(+/=/−)	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	12/0/0
Friedman rank	7.25	5.00	5.33	5.33	3.67	2.92	5.17	**1.33**
Rank	8	4	6	6	3	2	5	1

Table XI. Average computation time of RGBPOA and other algorithms

Dataset	BPOA	BASO	BSSA	BDE	BGWO	BHHO	BEO	RGBPOA
D(1)	5.4488	4.0501	4.3285	3.3644	**2.9374**	5.2249	4.0303	8.0461
D(2)	7.4822	6.3064	5.3132	5.4464	**3.7878**	8.2877	5.7948	7.6655
D(3)	4.6044	3.1156	3.2914	**2.4653**	2.5208	4.4822	3.0898	6.0975
D(4)	6.5941	6.4571	6.8683	4.7196	**3.8743**	7.6433	6.8468	11.5258
D(5)	4.7398	3.8398	4.0690	**2.5967**	2.7055	4.7117	3.7347	7.5008
D(6)	6.0750	5.3562	5.0789	4.6058	**3.3861**	7.2524	5.6512	8.4739
D(7)	4.8461	3.4012	3.5437	2.7673	**2.7009**	4.9224	3.4258	6.2600
D(8)	6.1294	4.6113	4.2489	4.6813	**3.1336**	7.2176	5.2271	7.0233
D(9)	4.7076	3.5611	3.7596	**2.5811**	2.6498	4.6649	3.4878	7.0223
D(10)	6.5109	5.0872	4.5484	4.8100	**3.2780**	7.5079	5.4144	7.5869
D(11)	5.5977	4.6863	4.6220	4.4016	**3.1562**	6.2510	5.2629	7.9423
D(12)	11.4014	8.3708	7.8391	7.7062	**5.1472**	12.4697	9.2274	11.4190
(+/=/−)	0/0/12	0/0/12	0/0/12	3/0/9	9/0/3	0/0/12	0/0/12	12/0/0
Friedman rank	6.17	3.83	3.83	2.08	**1.25**	6.92	4.17	7.75
Rank	6	3	3	2	1	7	5	8

Table XII. p-value of the Wilcoxon test classification accuracy of RGBPOA versus seven binary swarm intelligent optimization algorithms

Dataset	BPOA	BASO	BSSA	BDE	BGWO	BHHO	BEO
D(1)	4.56E-02	5.41E-03	8.79E-05	2.56E-07	2.91E-06	1.61E-07	3.08E-05
D(2)	2.03E-09	7.28E-04	8.99E-11	3.02E-11	3.02E-11	0.0451	2.60E-10
D(3)	5.61E-05	2.50E-03	3.37E-04	1.75E-05	6.76E-05	9.79E-05	6.76E-05
D(4)	2.31E-06	4.26E-05	3.01E-11	3.01E-11	3.01E-11	6.76E-05	3.01E-11
D(5)	1.56E-10	3.44E-10	1.10E-05	6.02E-11	2.01E-06	2.25E-08	7.13E-04
D(6)	1.57E-03	1.24E-02	1.60E-02	9.94E-07	4.84E-02	1.66E-04	5.91E-04
D(7)	1.42E-08	9.01E-03	2.73E-02	1.60E-10	1.76E-08	3.19E-06	5.08E-11
D(8)	4.56E-10	3.15E-05	2.99E-11	2.97E-11	3.00E-11	3.00E-11	2.98E-11
D(9)	4.98E-04	6.79E-03	7.69E-04	9.50E-06	4.98E-04	3.92E-02	9.49E-06
D(10)	5.46E-11	2.36E-10	1.20E-10	3.32E-11	4.94E-11	6.65E-11	3.00E-11
D(11)	3.19E-09	4.40E-03	3.01E-11	1.20E-10	1.87E-07	3.01E-11	6.05E-11
D(12)	2.31E-06	2.70E-03	2.49E-06	1.56E-08	6.35E-05	1.37E-03	7.73E-06

V. Conclusion and Future Works

This paper presents a novel approach to the feature selection (FS) problem through the introduction of the ReliefF-guided binary Pelican Optimization Algorithm (RGBPOA). To address the challenges associated with high-dimensional cancer gene expression data, the ReliefF-guided strategy was seamlessly integrated into the existing Pelican Optimization Algorithm (POA). This combination, along with techniques for feature addition and removal, was specifically designed to enhance classification accuracy. When applied to 12 diverse cancer gene expression datasets through two controlled experiments, RGBPOA demonstrated its capability to effectively reduce the number of features selected while improving classification performance and achieving lower fitness values in comparison to other methods. These promising results highlight the strength of RGBPOA, showcasing its superior performance over traditional swarm intelligence optimization algorithms that have been widely used in recent studies.

Moreover, this work opens up exciting future research directions in the realms of combinatorial optimization, heuristic algorithms, and feature selection. A potential avenue for further exploration is the development of new transfer functions for binary conversion, which could offer enhanced flexibility and precision in the feature selection process. Another promising area lies in the application of multi-objective swarm intelligence algorithms, which could further optimize both the efficiency and effectiveness of feature selection in complex, high-dimensional datasets. Additionally, combining advanced filtering techniques with packing algorithms could provide even more efficient solutions for identifying relevant feature subsets. By pursuing these directions, the field of feature selection will benefit from more robust and versatile algorithms, ultimately boosting the performance of machine learning models in environments characterized by high dimensionality and noise.

References

[1] F. Ahmad, N. A. M. Isa, Z. Hussain, and S. N. Sulaiman. “A Genetic Algorithm Based Multi-Objective Optimization of an Artificial Neural Network Classifier for Breast Cancer Diagnosis,” *Neural Computing & Applications*, vol. 23, no. 5, pp. 1427-1435, 2013.

[2] C. Tang, X. Zheng and W. Zhang. “Unsupervised Feature Selection via Multiple Graph Fusion and Feature Weight Learning,” *Science China Information Sciences*, vol. 66, no. 5, pp. 1-17, 2023.

[3] G. JagadeeswaraRao and A. Sivaprasad. “An Integrated Ensemble Learning Technique for Gene Expression Classification and Biomarker Identification from RNA-Seq Data for Pancreatic Cancer Prognosis,” *International Journal of Information Technology*, vol. 16, no. 3, pp. 1505-1516, 2024.

[4] J. Zhang and D. Xu. “FS-GBDT: Identification Multicancer-Risk Module via a Feature Selection Algorithm by Integrating Fisher Score and GBDT,” *Briefings in Bioinformatics*, vol. 22, no. 3, pp. bbaa189, 2021.

[5] M. Chen, K. Wu and B. Ni. “Searching the Search Space of Vision Transformer,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8714-8726, 2021.

[6] M. Ghasemi, M. A. Akbari and C. Jun. “Circulatory System Based Optimization (CSBO): An Expert Multilevel Biologically Inspired Meta-Heuristic Algorithm,” *Engineering Applications of Computational Fluid Mechanics*, vol. 16, no. 1, pp. 1483-1525, 2022.

[7] J. Zhou, X. Shen, and Y. Qiu. “Improving the Efficiency of Microseismic Source Locating Using a Heuristic Algorithm-Based Virtual Field Optimization Method,” *Geomechanics and Geophysics for Geo-Energy and Geo-Resources*, vol. 7, no. 3, pp. 89, 2021.

[8] W. Zhao, L. Wang, and Z. Zhang. “Atom Search Optimization and Its Application to Solve a Hydrogeologic Parameter Estimation Problem,” *Knowledge-Based Systems*, vol. 163, pp. 283-304, 2019.

[9] S. Mirjalili, A. H. Gandomi, and S. Z. Mirjalili. “Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems,” *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017.

[10] K. V. Price. “Differential Evolution,” *Handbook of Optimization: From Classical to Modern Approach*, pp. 187-214, 2013.

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis. “Grey Wolf Optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.

[12] A. A. Heidari, S. Mirjalili, and H. Faris. “Harris Hawks Optimization: Algorithm and Applications,” *Future Generation Computer Systems*, vol. 97, pp. 849-872, 2019.

[13] A. Faramarzi, M. Heidarinejad, and B. Stephens. “Equilibrium Optimizer: A Novel Optimization Algorithm,” *Knowledge-Based Systems*, vol. 191, pp. 105190, 2020.

[14] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili. “African Vultures Optimization Algorithm: A New Nature-Inspired Metaheuristic Algorithm for Global Optimization Problems,” *Computers & Industrial Engineering*, vol. 158, pp. 107408, 2021.

[15] P. Trojovský and M. Dehghani. “Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications,” *Sensors*, vol. 22, no. 3, pp. 855, 2022.

[16] D. Wang, X. Yang, and X. Liu. “Photoplethysmography-Based Blood Pressure Estimation Combining Filter-Wrapper Collaborated Feature Selection with LASSO-LSTM Model,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-14, 2021.

[17] A. Got, A. Moussaoui, and D. Zouache. “Hybrid Filter-Wrapper Feature Selection Using Whale Optimization Algorithm: A Multi-Objective Approach,” *Expert Systems with Applications*, vol. 183, pp. 115312, 2021.

[18] M. Zhang and J. S. Wang. “RG-NBEO: A ReliefF Guided Novel Binary Equilibrium Optimizer with Opposition-Based S-Shaped and V-Shaped Transfer Functions for Feature Selection,” *Artificial Intelligence Review*, vol. 56, no. 7, pp. 6509-6556, 2023.

[19] M. Jamei, A. Elbeltagi and S. Maroufpoor. “Combined Terrestrial Evapotranspiration Index Prediction Using a Hybrid Artificial Intelligence Paradigm Integrated with Relief Algorithm-Based Feature Selection,” *Computers and Electronics in Agriculture*, vol. 193, pp. 106687, 2022.

[20] J. Piri, P. Mohapatra and H. K. R. Singh. “An Enhanced Binary Multiobjective Hybrid Filter-Wrapper Chimp Optimization Based Feature Selection Method for COVID-19 Patient Health Prediction,” *IEEE Access*, vol. 10, pp. 100376-100396, 2022.

[21] X. Lu, W. He and Q. Lu. “Hybrid Filter-Wrapper Feature Selection Using Water Wave Optimization for Financial Crisis Prediction in Enterprises,” *International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 193-199, 2021.

[22] I. Kononenko. “Estimating Attributes: Analysis and Extensions of Relief,” *European Conference on Machine Learning*, pp. 171-182, 1994.

[23] N. S. Altman. “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,” *The American Statistician*, vol. 46, no. 3, pp. 175-185, 1992.