AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 56(2) (2024) 171-184
DOI: 10.22060/miscj.2024.23438.5376

Rank-based Adaptive Brooding in Mimetic Coral Reefs Search

Seyed Amirhossein Farjadi, Mehdi Alimohammadi, Mohammad-R. Akbarzadeh-T*

Department of Computer Engineering, Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Mashhad, Iran

ABSTRACT: Mimetic Coral Reefs Optimization (MCRO) has proven highly effective for feature selection due to its capacity to explore diverse solution spaces, enhancing model accuracy and robustness. However, integrating MCRO with local search techniques remains challenging, as it tends to be computationally intensive and prone to premature convergence. To address these issues, this paper introduces a Rank-based Adaptive Brooding (RAB) mechanism, designed to refine the local mimetic search strategy within MCRO. RAB adaptively adjusts the brooding operator based on the ranks of coral larvae, minimizing disruption to high-rank larvae and harnessing the exploratory potential of lower-rank larvae. This approach promotes a more balanced exploration-exploitation trade-off, leading to faster convergence and enhanced performance in complex problem spaces. The proposed method's efficacy is tested across eight UCI datasets using KNN, Decision Tree, and SVM classifiers, and the results are evaluated by precision, recall, and F1 score. Empirical results reveal that RAB outperforms existing adaptive strategies with fixed brooding, delivering superior feature selection performance, particularly in high-dimensional datasets. Additionally, the optimization capabilities of RAB were examined using 39 CEC benchmark functions, revealing consistent improvements in feature selection accuracy while demonstrating variable outcomes in broader optimization tasks. Notably, RAB showed significant enhancements in eight benchmark cases, highlighting its potential for broader applicability in optimization scenarios.

Review History:

Received: Aug. 14, 2024
Revised: Nov. 01, 2024
Accepted: Nov. 07, 2024
Available Online: Nov. 09, 2024

Keywords:

Feature Selection
Coral Reef Optimization
Rank-Based Brooding
Adaptation
Mimetic Search Strategy

1- Introduction

In high-dimensional data analysis, selecting feature subsets with notable and distinguishable effects is vital, especially in fields like genomics, medical image processing, and data mining, where high-dimensional datasets have gained prominence. However, the large dimensionality of these datasets can introduce irrelevant or redundant features, potentially limiting the effectiveness of learning algorithms or leading to data overfitting. To address these challenges, the RAB method was initially introduced to enhance feature selection accuracy within the Coral Reefs Optimization (CRO) algorithm. This approach leverages a ranking mechanism to adaptively guide the search process, aiming to improve the selection of relevant features.

Feature selection (FS) is a critical pre-processing step in machine learning and data mining to improve model performance by eliminating irrelevant and redundant features. The FS approach in [1] seeks to shorten the search space to enhance the effectiveness of the learning process by enhancing prediction and classification performance and shortening training time. Theng and Bhoyar [2] extensively survey feature selection techniques, emphasizing their role in improving decision-making quality. Similarly, Lung et al. [3] highlight the importance of feature selection in enhancing model accuracy and reducing complexity by removing unnecessary variables. Advances and challenges in feature selection methods are comprehensively reviewed by Ali et al. [4], showcasing their efficiency in handling high-dimensional datasets.

Furthermore, Chen et al. [5] demonstrate that feature selection, mainly using the Random Forest algorithm, significantly enhances classification accuracy and performance by eliminating unimportant variables and addressing the curse of dimensionality. Farag et al. [6] also comprehensively review feature selection and various optimization algorithms, emphasizing their crucial role in enhancing machine learning models across diverse scientific fields. These recent studies underscore feature selection's ongoing advancements and critical role in machine learning.

Recent advancements in feature selection algorithms have demonstrated significant improvements in handling high-dimensional datasets and enhancing classification accuracy. Kamala et al. [7] found that high-dimensional data can cause overfitting, and their Improved Hybrid Feature Selection (IHFS) method enhances prediction performance

*Corresponding author's email: akbarzadeh@ieee.org

by combining filter and wrapper techniques. Drotár et al. [8] studied ensemble feature selection methods using voting schemes and clustering, demonstrating improved robustness and performance across various accuracy measures. Pereira et al. [9] present a binary version of the Cuckoo Search algorithm (BCS) for feature selection, demonstrating its effectiveness compared to other nature-inspired optimization techniques in reducing noisy features and improving classification accuracy. Shikoun et al. [10] study the Binary Crayfish Optimization Algorithm (BinCOA) for feature selection, demonstrating its superior classification accuracy and feature reduction performance by incorporating novel enhancements like refracted opposition-based learning and crisscross strategies. The paper [11] investigates a new wrapper method called Binary Crow Search Algorithm (BCSA) for feature selection, demonstrating its effectiveness in improving classification accuracy and reducing computational cost compared to traditional methods. In [12], authors studied Hybrid Particle Swarm Optimization and Crow Search Algorithm with clustering initialization strategy (HPSOCSA-CIS) enhances feature selection by improving exploration and classification accuracy across various datasets.

In [13], the Binary Sailfish Optimizer (BSF) and its enhanced version with adaptive β-hill climbing (AβBSF) improve feature selection by effectively removing irrelevant features and outperforming other meta-heuristic methods on various datasets. Ahmed et al. [14] presented an improved Coral Reefs Optimizer with adaptive hill climbing for feature selection, demonstrating superior performance on 18 UCI datasets compared to 10 state-of-the-art methods. Xie et al. [15] research DENGO, an enhanced version of the Northern Goshawk Optimization algorithm, which improves feature selection by overcoming local optimum traps and slow convergence, demonstrating superior performance and stability compared to other methods. Numerous techniques, including Genetic Algorithm (GA) [16], Particle Swarm Optimization (PSO) [17], Differential Evolutionary [18], Ant Colony Optimization (ACO) [19], Scatter Search Algorithm (SSA) [20], Artificial Bee Colony (ABC) algorithm [21], Swallow Swarm Optimization (SSO) [22], Dragonfly Algorithm (DA) [23], and Archimedes Optimization Algorithm (AOA) [24], are helpful in the fields of optimization and feature selection.

The Mimetic Coral Reefs Optimization (MCRO) algorithm has emerged as a powerful metaheuristic for solving complex optimization problems, including feature selection. The integration of adaptive brooding within the MCRO framework is motivated by these existing methods' limitations, particularly in handling high-dimensional data. The proposed RAB mechanism prioritizes effective feature selection while aiming to overcome common issues like premature convergence by using rank-based adjustments that adapt to dataset complexity. Bérchez-Moreno et al. [25] explore novel memetic training for Artificial Neural Networks (ANNs) using Coral Reef Optimization algorithms, with the Dynamic Statistically-driven version (M-DSCRO) showing superior performance in classification accuracy and minority class performance compared to other methods. Salcedo-Sanz et al. [26] present the Coral Reefs Optimization algorithm (CRO) as a robust tool for solving complex optimization problems, demonstrating its applicability in real-world scenarios. Some years later, Salcedo-Sanz [27] reviews the latest developments in the Coral Reefs Optimization Algorithm, highlighting its effectiveness in various optimization scenarios. Durán-Rosal et al. [28] propose a novel modification of the CRO algorithm, called memetic CRO (MCRO), which effectively reduces the size of time series with minimal error, outperforming standard CRO and its variants in various applications. These sources collectively illustrate the versatility and efficacy of the CRO and MCRO algorithms in optimization tasks. RAB, inspired by natural coral reef ecosystems, enhances the MCRO algorithm by dynamically adjusting the selection pressure based on feature importance. This paper extends Farjadi and Akbarzadeh-T. [29] study, which integrated RAB in the MCRO algorithm, demonstrating its potential to improve feature selection outcomes.

The AβCRO algorithm, similar to other biologically [30] inspired meta-heuristics, incorporates an asexual mutation operator known as brooding to prevent premature convergence. Typically, this algorithm selects a portion of the population, determined by a fixed rate $\left(1-F_b\right)$, and the brooding operator is applied uniformly across these selected larvae. However, one drawback of using a constant brooding rate, irrespective of a larva's cost function, is that it causes highly fit larvae to undergo the same level of mutation as less fit larvae. Consequently, this can slow down the algorithm's convergence. To address this issue, this paper introduces a new approach that generates brooding probability based on the rank of each larva [31, 32]. The main contribution of this paper is to suggest Rank-Based Adaptive Brooding (RAB), in contrast to [14], which does not consider ranking mechanisms. We recommend applying a ranking mechanism that dynamically adjusts the brooding operator based on coral larvae ranks to prioritize exploration for lower-ranked larvae while preserving the characteristics of higher-ranked larvae. This would reduce disruption to well-performing solutions and enhance convergence speed.

In comparison, the [33] study applied standard CRO for feature selection, achieving high classification accuracy with specific classifiers. However, our RAB-enhanced approach introduces an adaptive mechanism that dynamically adjusts mutation rates based on solution rank, balancing exploration and exploitation. This advancement prevents premature convergence and boosts computational efficiency, demonstrating superior performance and broader applicability across various datasets and classifiers, particularly in handling high-dimensional data.

Yan et al. [34] integrate simulated annealing with CRO to enhance search performance for feature selection in high-dimensional biomedical datasets, the Rank-based Adaptive Brooding (RAB) method in our work focuses on dynamically adjusting the brooding operator based on cost function rankings. While BCROSAT enhances feature subset diversity using simulated annealing to escape local optima, our approach

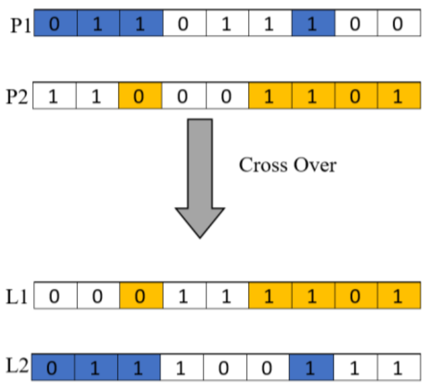

Fig. 1. Broadcast spawning step.

employs an adaptive rank-based brooding mechanism, balancing exploration and exploitation by prioritizing high-potential solutions. Unlike BCROSAT, which utilizes KNN exclusively, RAB was tested on various classifiers (SVM, Decision Tree, and KNN), providing broader insights into its generalizability and computational efficiency. Both methods address the "curse of dimensionality" in biomedical data. Still, RAB's adaptive approach to brooding demonstrates improvements in convergence speed and accuracy across diverse datasets, as shown in our results.

The structure of this paper is as follows: Section two offers a concise overview of the AβCRO algorithm, highlights the motivation behind employing the adaptive brooding operator, and introduces the concept of RAB. Section three details the experiments conducted to validate the proposed method and presents the results achieved by our approach, comparing them with other advanced solutions. Finally, Section four concludes the paper, discussing its limitations and proposing potential future research directions.

2- Method

2- 1- AβCRO Algorithm

In this algorithm, a coral larva represents a potential solution to the problem, competing with other corals to settle on the reefs and undergo growth and development. Integrating the memetic coral reefs algorithm with the adaptive beta hill-climbing search algorithm aims to avoid getting trapped in local optima. Overall, the AβCRO algorithm leverages the global search capabilities of the coral reefs optimization algorithm and the local search strengths of the AβHC search algorithm to identify the optimal feature subset.

The steps of this algorithm are as follows:

- **Initialization:** In this initial reef formation step, the algorithm begins by populating some grid squares in the problem space. A crucial parameter in the AβCRO algorithm is k_i, which, based on experiments conducted in this study, is set to 0.6, representing the ratio of occupied reefs to unoccupied ones.
- **Broadcast spawning (sexual external reproduction):** In this phase, a subset of corals is selected with a probability F_b to undergo broadcast spawning. The selected corals engage in sexual reproduction, resulting in the formation of new larvae (illustrated in Fig. 1).
- **Brooding (internal sexual reproduction):** In this stage, corals are selected with a probability of $(1-F_b)$, and the values of their larvae are altered randomly. These larvae are then released into the water, similar to the previous step (illustrated in Fig. 2).
- **Larvae settling:** Once all larvae at the stage k are formed through either broadcast spawning or brooding, they attempt to settle on the reef to grow. Each larva competes for space in the reef by trying to occupy a random square (i,j) in the grid. The coral can settle and grow if the square is vacant, regardless of its value. If a coral already occupies the square, the new larva can only replace it if it has a higher value. All larvae compete for space, and those with the highest values occupy the reef grid. Each larva has only a limited chance α to settle; otherwise, it will be displaced by more valuable larvae.
- **Budding or Fragmentation (Asexual reproduction):** In this phase, corals within the reef are arranged according to their cost function values. A fraction F_a of the coral population is then selected to produce a clone of itself.
- **AβHC Algorithm:** Hill Climbing (HC) is a straightforward local search algorithm, but its primary limitation is its tendency to get trapped in local optima, preventing it from finding global optima. To overcome this limitation, the βHC algorithm is introduced. However, βHC requires careful parameter tuning, which can be challenging and often requires extensive.
- Experimentation for each specific problem. To circumvent the need for such exhaustive experiments, an adaptive model called AβHC is proposed. This model repeatedly

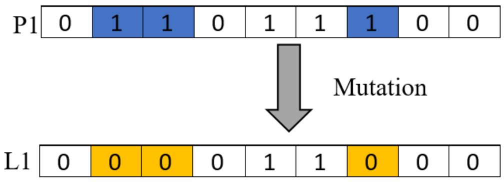

Fig. 2. The brooding process (color variations indicate larvae that have undergone mutation).

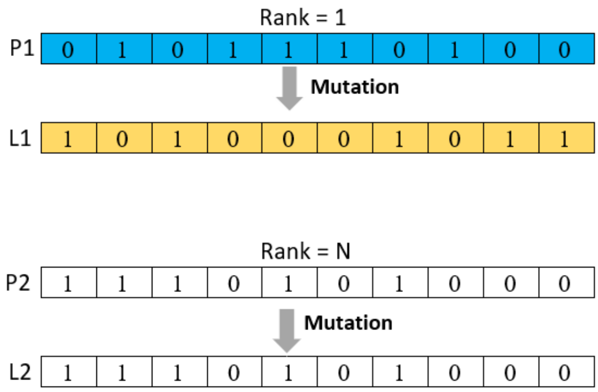

Fig. 3. Ranked-based adaptive brooding mechanism (color changes indicate larvae that have undergone mutation).

refines the solution generated by the coral reefs algorithm using two operators: the N-operator and the β-operator.

- **Depredation:** During each iteration, a small portion of the coral population is depredated, where weaker corals are replaced by stronger ones based on their cost function values. This process frees up space for new corals to settle.
- **Cost function evaluation:** To assess and compare the effectiveness of different solutions, we calculate their cost function values using a specific formula (referenced as equation 1). This cost function value helps identify the best feature subset. Where ω denotes the weightage given to the classification error, $(1-A)$ represents the classification error, and $\left(\frac{d}{D}\right)$ represents the fraction of features selected from the original feature set.

Objective function :

$$f(A,d) = \omega \,.\, (1-A) + (1-\omega) \,.\, \frac{d}{D} \tag{1}$$

2- 2- The Motivation Behind Adaptive Brooding

Premature convergence in the AβCRO algorithm can be attributed to two main factors: insufficient genetic diversity in the initial population and the loss of genetic information during the optimization process. The brooding mechanism is crucial in exploring uncharted problem spaces, generating new genetic information, and recovering lost data. In the standard AβCRO, brooding occurs with a constant probability across all individuals. While a higher brooding probability enhances exploration, it can also result in the loss of valuable information from above-average individuals, ultimately leading to suboptimal convergence.

To address this, adaptive brooding mutates above-average individuals with a very low probability and below-average individuals with a higher probability [35, 36].

2- 3- Ranked-Based Adaptive Brooding: the proposed method

The primary aim of this research is to advance feature selection accuracy by developing an adaptive brooding mechanism that ranks coral larvae based on cost function, thus enabling the identification of highly relevant feature subsets. The proposed method filters out irrelevant or redundant features by effectively managing high-dimensional datasets, significantly improving computational efficiency. Furthermore, by dynamically adjusting the brooding probability based on larval rank, the approach ensures a balanced exploration-exploitation trade-off, reducing the risk of premature convergence and enhancing convergence speed. These innovations elevate existing CRO-based approaches and pave the way for a more versatile and adaptive optimization tool. This method holds potential for broader applications across various complex optimization challenges, setting a foundation for further refinement and adaptability in diverse data-driven fields.

In this approach, each larva's rank is determined by its relative cost function within the population. The fittest larva is ranked N in a population of N individuals, while the lowest fitness is ranked 1. The remaining individuals are ranked between 1 and N based on cost function values. The normalized rank is then used to calculate the brooding probability, as described in Equation (2).

$$p = p_{MAX} * (1 - \frac{r-1}{N-1}) \tag{2}$$

The brooding probability of a larva, denoted by p , is determined by the RAB mechanism, where p_{MAX} is the maximum brooding probability, r represents the larva's rank, and N indicates the population size. Equation (1) is designed to ensure that the best-performing larva has a brooding probability of zero, while the least-performing larva has the highest probability, p_{MAX} . The brooding probabilities for other larvae are distributed linearly between 0 and p_{MAX} according to their ranks. Figure 3 illustrates the concept of the adaptive brooding operator, and Figure 4 shows the different phases of the AβCRO algorithm, which were discussed in section two.

3- Results and discussion

Three classifiers—Decision Tree, SVM, and KNN [37]—were used to evaluate the classification accuracy of the feature subsets selected by the proposed FS model. Following the methodology in [30, 38, 39], the dataset was divided into two parts: 80% was used for training and classification, and 20% was reserved for testing. The RAB mechanism aims to improve feature selection through an adaptive approach, enabling better classification performance and faster convergence.

3- 1- Software

The experiments were conducted on a system equipped with an Intel® Pentium® G2020 processor and 7.6 GB of RAM. Each dataset was run 15 times, and the best result was selected for further analysis. Table 1 provides the execution time in seconds for each dataset using AβCRO and Rank-Based AβCRO for a single run.

3- 2- Dataset Description

Eight standard UCI datasets were utilized to evaluate the performance of AβCRO and Rank-Based AβCRO, encompassing a range of domains. Since these datasets in [40] did not achieve 100% accuracy, this study aims to improve their accuracy by applying the RAB concept.

The datasets include a mixture of binary and multi-class classifications with varying numbers of features, providing a comprehensive basis to assess the generalizability of the proposed method.

The execution times in Table 1 indicate that the Rank-Based AβCRO consistently converges faster than AβCRO, supporting the claim that the adaptive brooding method improves computational efficiency. This efficiency is attributed to the adaptive nature of RAB, which prioritizes

Table 1. Execution time(in seconds) for a single Run on eight stand UCI datasets using ABCRO and ranked-based ABCRO.

S1. No.	Dataset	AβCRO (sec)	Ranked based AβCRO (sec)
1	Breastcancer	19	12.46
2	Tic-Tac-Toe	9.79	9.23
3	HeartEw	18	10
4	Exactly2	8.67	8.41
5	SpectEW	9.21	8.68
6	IonosphereEW	15.71	14
7	KrvskEW	14.8	14.24
8	WaveformEW	140	108

Table 2. Evaluation of AβCRO and Ranked-Based AβCRO in terms of Precision, Recall, and F1 Score across eight UCI datasets (Best results are highlighted in bold).

S1. no.	Dataset	AβCRO			Ranked Based AβCRO		
		Precision	*Recall*	*f1 score*	*Precision*	*Recall*	*f1 score*
1	Breast cancer	**1**	**1**	**1**	**1**	**1**	**1**
2	Tic-Tac-Toe	0.87	0.9	0.83	**0.89**	**0.92**	**0.85**
3	HeartEW	0.92	0.86	0.85	**0.945**	**0.875**	**0.855**
4	Exactly2	0.42	0.67	0.312	**0.423**	**0.71**	**0.42**
5	SpectEW	0.552	0.68	0.61	**0.68**	**0.73**	**0.73**
6	IonosphereEW	0.971	0.98	0.92	**0.985**	**0.992**	**0.964**
7	KrvskpEW	0.982	0.983	0.99	**0.997**	**0.998**	**1**

high-potential solutions early on, reducing unnecessary computational overhead.

3- 3- Statistical Analysis

To assess the effectiveness of Rank-Based AβCRO, statistical metrics such as Precision, Recall, and F1 score were evaluated across eight datasets, as shown in Table 2. These metrics provide insights into each dataset's accuracy and robustness and demonstrate the proposed method's superiority.

3- 4- Result Analysis

This section details the outcomes of the proposed feature selection method, Ranked-Based AβCRO. The performance of this algorithm was assessed using Decision Tree, SVM, and KNN classifiers, as shown in Table 3.

The results demonstrate the enhanced effectiveness of Ranked-Based AβCRO in identifying better results than the standard AβCRO algorithm, primarily due to the adaptive brooding mechanism's ability to prioritize relevant features dynamically. The Ranked-Based AβCRO algorithm surpasses expectations with an SVM classifier across the UCI datasets (using Gaussian SVM with $\gamma = 2$ and $C = 1$). This improvement is likely due to SVM's sensitivity to well-selected features, as the adaptive brooding process emphasizes selecting only the most influential attributes, thus minimizing noise and improving classification accuracy.

3- 4- 1- Classifier Performance Comparison

Table 3 highlights the efficiency of the Ranked-Based AβCRO with different classifiers:

- **Decision Tree Classifier:** Ranked-Based AβCRO with the Decision Tree classifier outperforms AβCRO in two datasets, selecting fewer features in four datasets. This observation suggests that the method's adaptability aligns well with the Decision Tree's need for clear, relevant features, reducing overfitting and maintaining high classification accuracy.
- **SVM Classifier:** Ranked-Based AβCRO with the SVM classifier demonstrates the highest overall accuracy, outperforming AβCRO in six datasets, selecting fewer features in three datasets, and matching performance in the Breastcancer dataset.
- The SVM classifier benefits from the RAB method's emphasis on feature relevance, as fewer, more critical features yield enhanced model precision. Notably, except for the KrvskpEW dataset, this classifier has a shorter execution time across all datasets, indicating computational efficiency.
- **KNN Classifier:** Ranked-Based AβCRO achieves better accuracy in three datasets and fewer features in five datasets with the KNN classifier. This performance shows that the RAB approach helps manage KNN's sensitivity to irrelevant features, allowing it to perform well in datasets with complex feature spaces.

3- 4- 2- Execution Time

As shown in Table 3, the Ranked-Based AβCRO algorithm consistently provides faster convergence than standard AβCRO:

The Decision Tree classifier converges the fastest, followed by KNN, with SVM demonstrating slightly longer but still efficient execution times across most datasets. This efficiency can be attributed to the RAB approach's focused

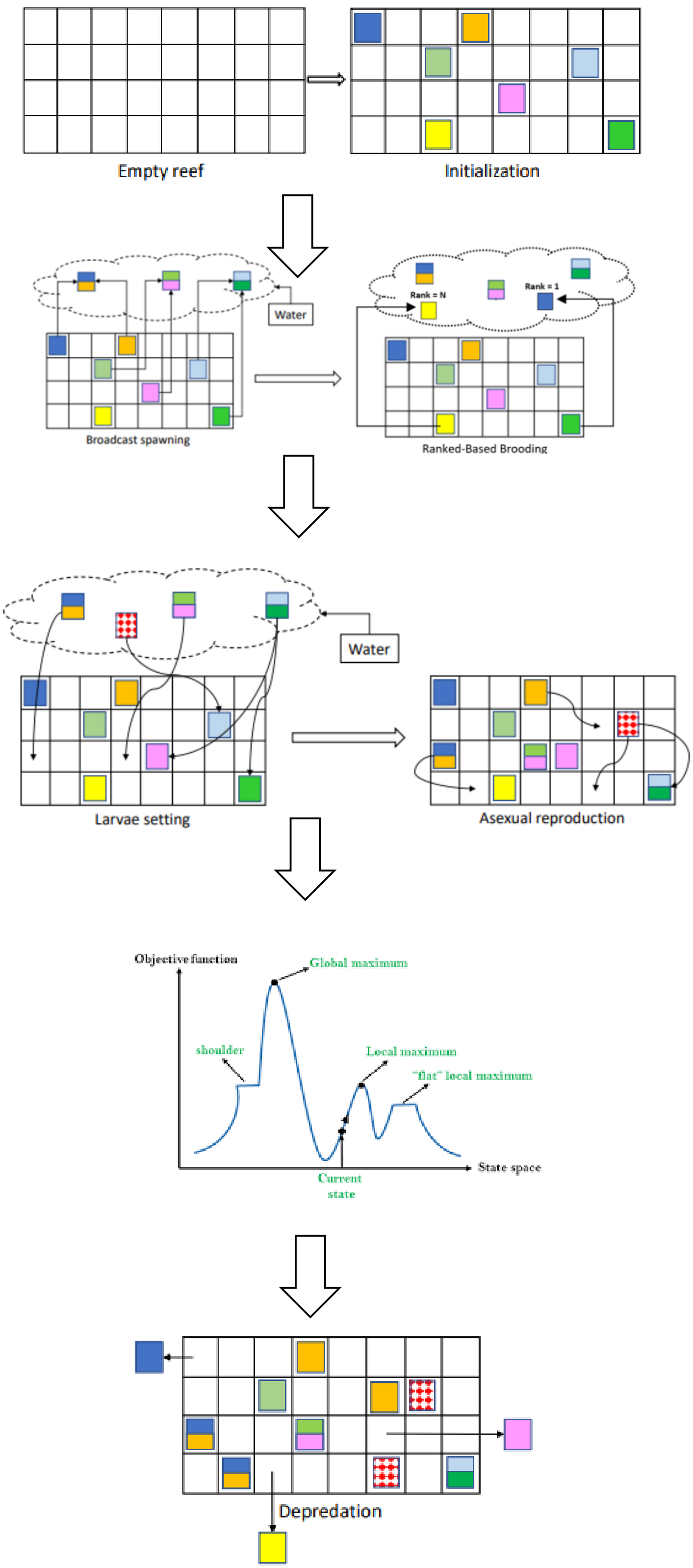

Fig. 4. The key phases of the ranked-based AβCRO algorithm.

Table 3. Performance of ABCRO and Ranked Based ABCRO in terms of Classification Accuracy, Selected Features, and Execution time using Decision Tree, SVM, and KNN Classifiers (Highest Classification Accuracy, Lowest No. of Selected Features, and Shortest Execution time are highlighted).

Sl. no.	Dataset	Classifier	AβCRO			Ranked Based AβCRO		
			Accuracy	*Features*	*Execution Time (s)*	*Accuracy*	*Features*	*Execution Time (s)*
1	Breast cancer	Decision Tree	**100**	3	151	98.57	**2**	**141**
2	Tic-Tac-Toe		83.85	**6**	138	**84.37**	7	**132**
3	HeartEW		**94.44**	6	145	**94.44**	**3**	**136**
4	Exactly2		**76**	**1**	137	**76**	**1**	**130**
5	SpectEW		**96.29**	5	144	95	**4**	**136**
6	IonosphereEW		98.57	**4**	227	**100**	5	**215**
7	KrvskpEW		**95.77**	4	226	95.46	**3**	**218**
8	WaveformEW		**79.7**	**6**	1663	79.5	7	**1593**
9	Breastcancer	SVM	**99.28**	**2**	784	98.57	**2**	**777**
10	Tic-Tac-Toe		85.93	6	1685	**90.1**	**5**	**1600**
11	HeartEW		**100**	**3**	315	98.14	5	**306**
12	Exactly2		76	**1**	1284	**85**	11	**1232**
13	SpectEW		94.44	**5**	274	**96.29**	6	**255**
14	IonosphereEW		100	**4**	420	**100**	**4**	**406**
15	KrvskpEW		98.9	**13**	**10542**	**99.21**	14	10702
16	WaveformEW		87.2	17	12021	**88**	**15**	**11011**
17	Breastcancer	KNN	**100**	3	1137	98.57	**2**	**1001**
18	Tic-Tac-Toe		84.89	**9**	1508	**86.45**	**9**	**1305**
19	HeartEW		**96.29**	4	567	90.74	**3**	**542**
20	Exactly2		**76**	**1**	1525	**76**	**1**	**1400**
21	SpectEW		**96.29**	8	601	94.44	**4**	**541**
22	IonosphereEW		**100**	6	979	98.57	**5**	**840**
23	KrvskpEW		98.27	**9**	8102	**98.74**	15	**7501**
24	WaveformEW		**87.1**	18	13037	87	**14**	**12476**

exploration-exploitation balance, which minimizes the computational effort required to reach optimal solutions.

The Ranked-Based AβCRO surpasses AβCRO in execution time across all eight datasets. This reduction in computation time is particularly notable in large datasets, such as WaveformEW, where RAB achieved a 20% reduction in processing time. This efficiency is likely due to RAB's adaptive feature selection process, which streamlines the data for classifiers, reducing the complexity of training.

3- 4- 3- Feature Reduction Analysis

- The Ranked-Based AβCRO algorithm consistently selects fewer features without sacrificing accuracy, validating the effectiveness of its feature selection process:
- High-dimensional datasets (such as IonosphereEW and WaveformEW) demonstrate significant feature reduction while maintaining or improving accuracy. For example, the Ranked-Based AβCRO on SpectEW selects one fewer feature than AβCRO, with similar or improved accuracy, showing that the adaptive brooding mechanism effectively minimizes redundant features.
- This feature reduction is crucial for classifiers sensitive to overfitting (e.g., Decision Tree), as fewer features lead to simpler models with reduced risk of overfitting.

These findings support the RAB method's suitability for feature selection in high-dimensional datasets, especially for applications where computational efficiency and accuracy are priorities.

3- 5- Benchmark Optimization Results

To evaluate the RAB method's broader optimization capabilities, we applied it to benchmark functions from CEC2017 and CEC2021. Tables 4 and 5 display the results across different configurations of Adaptive Coral Reef Optimization (A-CRO) variants compared to the original CRO algorithm.

3- 6- Optimization Performance Analysis

The experimental results indicate that while the RAB method exhibits improvements on specific benchmarks, its overall optimization performance is variable:

- In specific benchmarks, RAB achieves marginal improvements. However, some functions display less promising outcomes, highlighting the complexity of parameter tuning in these optimization contexts.
- We expect the results to not be significantly better due to the different characteristics between feature selection and general optimization benchmarks. They might not be optimal with limited function evaluations. Additionally, we need to tune parameters due to their sensitivity for general optimization benchmarks to find optimal results.

3- 7- Parameter Sensitivity and Convergence

Figure 5 shows the cost function reduction for each dataset during the simulation using DT as a classifier, and the convergence speed can be seen for each one. In some cases, the proposed method converges faster than the A β CRO.

The RAB method demonstrates sensitivity to parameter configurations, particularly in convergence behavior:

- **Convergence Speed:** RAB converges faster on high-precision tasks, where the adaptive brooding mechanism directs focus to promising solutions early, minimizing exploration time.
- **Parameter Tuning:** Different settings, such as the brooding rate and population size, significantly impact the method's optimization performance. Fine-tuning these parameters may further harness RAB's potential in general optimization tasks, as the current configurations show variability across different test functions.

While Ranked-Based AβCRO achieves superior classification performance and computational efficiency in feature selection, further research is recommended to refine its application in broader optimization tasks.

The findings indicate that the adaptive nature of the RAB method provides a strong foundation for high-dimensional feature selection. Yet, parameter refinement may be necessary for fully leveraging its capabilities in diverse optimization problems.

4- Conclusion

This study highlights the crucial role of the RAB method in enhancing the Coral Reefs Optimization (CRO) algorithm, particularly in addressing complex optimization challenges. We introduced an adaptive mechanism to improve the algorithm's effectiveness across various scenarios, aiming to broaden its applicability.

Through this approach, we directly address intricate problems, demonstrating the need for adaptive strategies in evolutionary algorithms. By focusing on these enhancements, we aim to make CRO a more versatile tool for optimization. The adaptive brooding mechanism not only enhances classification accuracy but also improves the quality of convergence by effectively integrating the AβCRO approach. Our findings show that specific configurations of the RAB method yield marginal improvements in specific benchmarks, revealing the method's potential under particular conditions.

However, the overall performance varies, illustrating the inherent challenges in parameter tuning and the method's sensitivity to different optimization tasks. These results indicate that while the method shows promise, its effectiveness heavily depends on the specific characteristics of the problem at hand. This variability highlights the importance of ongoing experimentation and refinement to achieve consistent results.

In future work, we will prioritize refining the adaptive brooding approach to enhance its robustness and reliability across a broader range of optimization tasks. We plan to explore hybrid strategies that combine the strengths of RAB with other optimization techniques to boost performance. We aim to develop a more comprehensive and practical algorithm that consistently delivers superior results by addressing current limitations. This ongoing effort will strengthen the CRO algorithm, making it a more robust and adaptable tool for solving complex optimization problems.

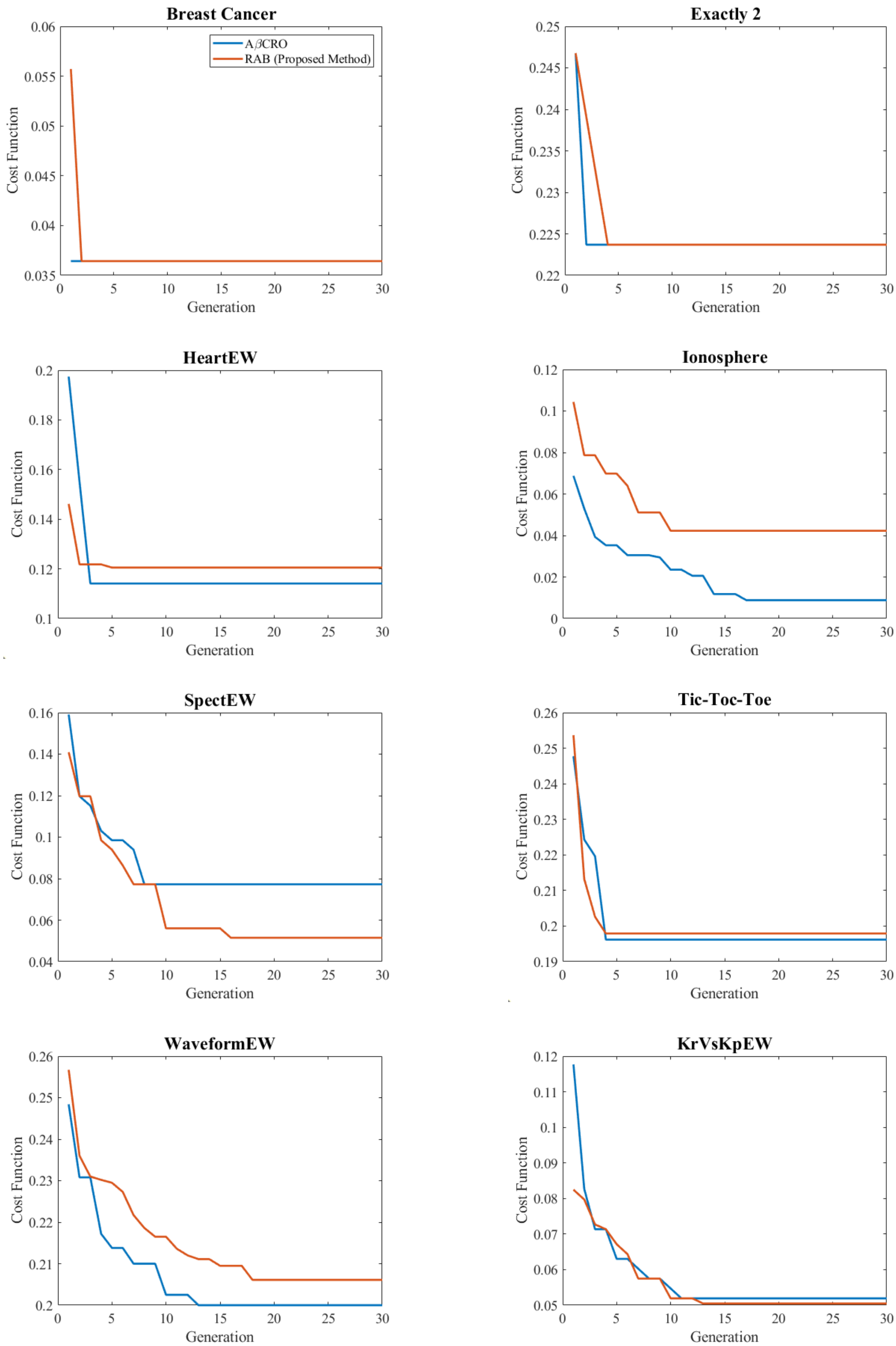

Fig. 5. The key phases of the ranked-based AβCRO algorithm.

Table 4. Configuration Parameters for Adaptive Coral Reef Optimization (A-CRO) Variants

Algorithm	Epoch	Population Size	po	Fb	Fa	Fd	Pd	GCR	gamma_min	gamma_max	n_trials
A-CRO1	1500	50	0.6	0.8	0.1	0.1	0.1	0.1	0.02	0.2	3
A-CRO2	1500	50	0.4	0.9	0.1	0.1	0.5	0.1	0.02	0.2	5
A-CRO3	1500	50	0.3	0.9	0.1	0.1	0.1	0.1	0.02	0.2	3
A-CRO4	1500	50	0.8	0.98	0.1	0.1	0.4	0.8	0.02	0.2	3

Table 5. Results for CEC2017.

Function	A-CRO1	A-CRO2	A-CRO3	A-CRO4	CRO1	CRO2	CRO3	CRO4
F1	9.43E+04	1.87E+05	1.37E+05	8.70E+06	**1.70E+04**	4.86E+04	4.93E+04	9.38E+06
F2	4.51E-01	8.03E-01	5.65E-01	1.42E+01	**2.38E-02**	9.85E-02	1.07E-01	2.20E+01
F3	1.38E+01	1.07E+01	1.64E+01	**9.65E+00**	1.65E+01	1.20E+01	1.22E+01	2.08E+01
F4	2.93E+01	3.46E+01	2.75E+01	7.26E+01	2.06E+01	2.35E+01	**2.03E+01**	8.29E+01
F5	9.35E-05	**6.58E-06**	3.63E-05	1.84E-04	5.59E-05	2.98E-05	7.17E-05	2.34E-04
F6	6.03E+01	6.48E+01	6.08E+01	2.99E+02	**4.33E+01**	5.39E+01	5.24E+01	3.65E+02
F7	6.74E+01	6.49E+01	6.61E+01	6.77E+01	**4.50E+01**	6.53E+01	6.00E+01	6.99E+01
F8	8.33E-01	1.41E+00	9.53E-01	**4.96E-01**	7.37E-01	8.13E-01	6.09E-01	1.51E+00
F9	7.13E+02	6.90E+02	7.03E+02	9.13E+02	**5.98E+02**	6.17E+02	6.36E+02	9.71E+02
F10	5.20E+03	3.82E+03	2.15E+03	1.00E+04	2.76E+03	**1.87E+03**	3.13E+03	1.32E+04
F11	3.51E+04	3.11E+04	3.53E+04	3.45E+06	**2.14E+04**	2.20E+04	3.42E+04	5.37E+06
F12	5.14E+03	5.07E+03	5.49E+03	6.71E+04	**3.17E+03**	4.64E+03	6.31E+03	7.71E+04
F13	1.11E+04	**8.69E+03**	1.09E+04	1.44E+04	1.25E+04	1.11E+04	1.04E+04	1.83E+04
F14	2.20E+04	1.32E+04	2.38E+04	6.89E+04	9.82E+03	**8.83E+03**	1.18E+04	3.08E+05
F15	5.03E+03	4.99E+03	2.80E+03	**2.69E+03**	3.81E+03	3.91E+03	4.12E+03	2.87E+03
F16	2.97E+02	3.61E+02	5.07E+02	5.17E+02	**2.75E+02**	3.95E+02	2.96E+02	6.50E+02
F17	**6.13E+03**	9.10E+03	8.22E+03	1.62E+04	8.22E+03	8.49E+03	7.54E+03	1.73E+04
F18	6.70E+03	5.97E+03	5.38E+03	6.02E+03	6.00E+03	**4.82E+03**	6.78E+03	1.22E+04
F19	1.11E+02	8.99E+01	6.63E+01	1.44E+02	9.47E+01	7.52E+01	**5.17E+01**	1.08E+02
F20	1.31E+02	**1.08E+02**	1.30E+02	1.49E+02	1.18E+02	1.15E+02	1.37E+02	1.74E+02
F21	1.04E+02	1.05E+02	1.05E+02	1.09E+02	1.05E+02	**1.04E+02**	1.05E+02	1.10E+02
F22	1.48E+02	1.60E+02	1.56E+02	2.20E+02	1.39E+02	**1.36E+02**	1.47E+02	2.27E+02
F23	2.19E+02	1.99E+02	2.12E+02	3.33E+02	2.19E+02	**1.98E+02**	1.99E+02	3.32E+02
F24	5.14E+02	5.06E+02	5.04E+02	5.09E+02	**4.98E+02**	5.09E+02	5.11E+02	5.04E+02
F25	4.86E+02	4.83E+02	4.70E+02	4.77E+02	4.74E+02	**4.50E+02**	4.74E+02	4.59E+02
F26	**3.78E+02**	4.23E+02	4.09E+02	4.11E+02	3.99E+02	4.07E+02	4.08E+02	4.09E+02
F27	1.29E+02	1.08E+02	1.12E+02	9.28E+01	**4.27E+01**	1.03E+02	9.05E+01	1.47E+02
F28	9.28E+03	1.13E+04	1.19E+04	7.69E+03	**4.84E+03**	5.78E+03	8.21E+03	1.36E+04
F29	1.86E+05	8.34E+05	1.83E+05	1.62E+05	1.21E+05	**6.32E+04**	2.15E+05	5.54E+05

Table 6. Results for CEC2021.

Function	A-CRO1	A-CRO2	A-CRO3	A-CRO4	CRO1	CRO2	CRO3	CRO4
F1	1.04E+05	1.77E+05	1.77E+05	6.39E+06	**1.59E+04**	5.52E+05	3.46E+04	8.75E+06
F2	6.53E+02	7.71E+02	7.71E+02	8.54E+02	5.89E+02	**4.68E+02**	6.57E+02	1.01E+03
F3	6.48E+01	6.55E+01	6.55E+01	3.00E+02	**4.62E+01**	9.71E+01	5.83E+01	3.86E+02
F4	4.38E+00	3.91E+00	3.91E+00	3.93E+00	**2.33E+00**	3.30E+00	2.91E+00	5.12E+00
F5	1.16E+04	1.41E+04	1.41E+04	2.14E+04	**7.68E+03**	9.65E+03	8.37E+03	3.44E+04
F6	2.78E+03	2.63E+03	2.63E+03	2.69E+03	**2.41E+03**	2.89E+03	4.45E+03	3.24E+03
F7	9.77E+02	6.47E+02	6.47E+02	4.55E+03	**3.63E+02**	4.51E+03	7.48E+02	9.06E+03
F8	1.05E+02	1.05E+02	1.05E+02	1.10E+02	**1.05E+02**	1.05E+02	1.05E+02	1.11E+02
F9	2.09E+02	2.11E+02	2.11E+02	3.10E+02	2.05E+02	2.24E+02	**1.98E+02**	3.24E+02
F10	5.04E+02	5.17E+02	5.17E+02	5.07E+02	**5.00E+02**	5.03E+02	5.02E+02	5.02E+02

References

[1] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," Neurocomputing, vol. 300, pp. 70-79, 2018.

[2] D. Theng and K. K. Bhoyar, "Feature selection techniques for machine learning: a survey of more than two decades of research," Knowledge and Information Systems, vol. 66, no. 3, pp. 1575-1637, 2024.

[3] R. I. Lung and M.-A. Suciu, "An Evolutionary Approach to Feature Selection and Classification," in International Conference on Machine Learning, Optimization, and Data Science, 2023: Springer, pp. 333-347.

[4] M. Z. Ali, A. Abdullah, A. M. Zaki, F. H. Rizk, M. M. Eid, and E. M. El-Kenway, "Advances and challenges in feature selection methods: a comprehensive review," J Artif Intell Metaheuristics, vol. 7, no. 1, pp. 67-77, 2024.

[5] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," Journal of Big Data, vol. 7, no. 1, p. 52, 2020.

[6] A. A. Farag, Z. M. Ali, A. M. Zaki, F. H. Rizk, M. M. Eid, and E.-S. M. EL-Kenawy, "Exploring Optimization Algorithms: A Review of Methods and Applications," Full Length Article, vol. 7, no. 2, pp. 08-8-17, 2024.

[7] R. Kamala and R. J. Thangaiah, "An improved hybrid feature selection method for huge dimensional datasets," IAES International Journal of Artificial Intelligence, vol. 8, no. 1, p. 77, 2019.

[8] P. Drotár, M. Gazda, and L. Vokorokos, "Ensemble feature selection using election methods and ranker clustering," Information Sciences, vol. 480, pp. 365-380, 2019.

[9] L. Pereira et al., "A binary cuckoo search and its application for feature selection," Cuckoo Search and Firefly Algorithm: Theory and Applications, pp. 141-154, 2014.

[10] N. H. Shikoun, A. S. Al-Eraqi, and I. S. Fathi, "BinCOA: An Efficient Binary Crayfish Optimization Algorithm for Feature Selection," IEEE Access, vol. 12, pp. 28621-28635, 2024.

[11] R. C. T. De Souza, L. dos Santos Coelho, C. A. De Macedo, and J. Pierezan, "A V-shaped binary crow search algorithm for feature selection," in 2018 IEEE Congress on Evolutionary computation (CEC), 2018: IEEE, pp. 1-8.

[12] J. Osei-Kwakye, F. Han, A. A. Amponsah, Q.-H. Ling, and T. A. Abeo, "A diversity enhanced hybrid particle swarm optimization and crow search algorithm for feature selection," Applied Intelligence, vol. 53, no. 17, pp. 20535-20560, 2023.

[13] K. K. Ghosh, S. Ahmed, P. K. Singh, Z. W. Geem, and R. Sarkar, "Improved binary sailfish optimizer based on adaptive β-hill climbing for feature selection," IEEE access, vol. 8, pp. 83548-83560, 2020.

[14] S. Ahmed, K. K. Ghosh, L. Garcia-Hernandez, A. Abraham, and R. Sarkar, "Improved coral reefs optimization with adaptive β-hill climbing for feature selection," Neural Computing and Applications, vol. 33, no. 12, pp. 6467-6486, 2021.

[15] R. Xie, S. Li, and F. Wu, "An Improved Northern Goshawk Optimization Algorithm for Feature Selection," Journal of Bionic Engineering, pp. 1-39, 2024.

[16] S. Mirjalili, “Evolutionary algorithms and neural networks,” Studies in computational intelligence, vol. 780, pp. 43-53, 2019.

[17] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proceedings of ICNN’95-international conference on neural networks, 1995, vol. 4: ieee, pp. 1942-1948.

[18] 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.

[19] C. C. Ribeiro, P. Hansen, V. Maniezzo, and A. Carbonaro, “Ant colony optimization: an overview,” Essays and surveys in metaheuristics, pp. 469-492, 2002.

[20] F. C. García López, M. García Torres, J. A. Moreno Pérez, and J. M. Moreno Vega, “Scatter search for the feature selection problem,” in Conference on Technology Transfer, 2003: Springer, pp. 517-525.

[21] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” Journal of global optimization, vol. 39, pp. 459-471, 2007.

[22] M. Neshat, G. Sepidnam, and M. Sargolzaei, “Swallow swarm optimization algorithm: a new method to optimization,” Neural Computing and Applications, vol. 23, no. 2, pp. 429-454, 2013.

[23] S. Mirjalili, “Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,” Neural computing and applications, vol. 27, pp. 1053-1073, 2016.

[24] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, “Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems,” Applied Intelligence, vol. 51, pp. 1531-1551, 2021.

[25] F. Bérchez-Moreno, A. M. Durán-Rosal, C. Hervás Martínez, P. A. Gutiérrez, and J. C. Fernández, “A memetic dynamic coral reef optimisation algorithm for simultaneous training, design, and optimisation of artificial neural networks,” Scientific Reports, vol. 14, no. 1, p. 6961, 2024.

[26] S. Salcedo-Sanz, J. Del Ser, I. Landa-Torres, S. Gil-López, and J. Portilla-Figueras, “The coral reefs optimization algorithm: a novel metaheuristic for efficiently solving optimization problems,” The Scientific World Journal, vol. 2014, no. 1, p. 739768, 2014.

[27] S. Salcedo-Sanz, “A review on the coral reefs optimization algorithm: new development lines and current applications,” Progress in Artificial Intelligence, vol. 6, pp. 1-15, 2017.

[28] A. M. Durán-Rosal, P. A. Gutiérrez, S. Salcedo-Sanz, and C. Hervás-Martínez, “An empirical validation of a new memetic CRO algorithm for the approximation of time series,” in Conference of the Spanish Association for Artificial Intelligence, 2018: Springer, pp. 209-218.

[29] S. A. Farjadi and M.-R. Akbarzadeh-T, “Rank-Based Adaptive Brooding in a Mimetic Coral Reefs Search for Feature Selection,” in 2023 31st International Conference on Electrical Engineering (ICEE), 2023: IEEE, pp. 177-182.

[30] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, “Efficient hybrid nature-inspired binary optimizers for feature selection,” Cognitive Computation, vol. 12, no. 1, pp. 150-175, 2020.

[31] M. Srinivas and L. M. Patnaik, “Adaptive probabilities of crossover and mutation in genetic algorithms,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 4, pp. 656-667, 1994.

[32] I. A. Korejo, Z. Khuhro, F. Jokhio, N. Channa, and H. Nizamani, “An adaptive crossover operator for genetic algorithms to solve the optimization problems,” Sindh University Research Journal-SURJ (Science Series), vol. 45, no. 2, 2013.

[33] L. M. Abouelmagd, M. Y. Shams, N. E. El-Attar, and A. E. Hassanien, “Feature selection based coral reefs optimization for breast cancer classification,” in Medical Informatics and Bioimaging Using Artificial Intelligence: Challenges, Issues, Innovations and Recent Developments: Springer, 2021, pp. 53-72.

[34] C. Yan, J. Ma, H. Luo, and A. Patel, “Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets,” Chemometrics and Intelligent Laboratory Systems, vol. 184, pp. 102-111, 2019.

[35] B. Rajakumar and A. George, “A new adaptive mutation technique for genetic algorithm,” in 2012 IEEE International Conference on Computational Intelligence and Computing Research, 2012: IEEE, pp. 1-7.

[36] D. Thierens, “Adaptive mutation rate control schemes in genetic algorithms,” in Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02 (Cat. No. 02TH8600), 2002, vol. 1: IEEE, pp. 980-985.

[37] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” The American Statistician, vol. 46, no. 3, pp. 175-185, 1992.

[38] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary grey wolf optimization approaches for feature selection,” Neurocomputing, vol. 172, pp. 371-381, 2016.

[39] M. M. Mafarja and S. Mirjalili, “Hybrid whale optimization algorithm with simulated annealing for feature selection,” Neurocomputing, vol. 260, pp. 302-312, 2017.

[40] S. Ahmed, K. K. Ghosh, L. Garcia-Hernandez, A. Abraham, and R. Sarkar, “Improved coral reefs optimization with adaptive β-hill climbing for feature selection,” Neural Computing and Applications, vol. 33, no. 12, pp. 6467-6486, 2021.

HOW TO CITE THIS ARTICLE
S. A. H. Farjadi, M. Alimohammadi, M. R. Akbarzadeh-T, Rank-based Adaptive Brooding in Mimetic Coral Reefs Search, AUT J. Model. Simul., 56(2) (2024) 171-184.

DOI: 10.22060/miscj.2024.23438.5376