Expert Systems With Applications 271 (2025) 126618

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Attribute reduction based on adaptive neighborhood rough sets and three-way pied kingfisher optimizer

Wenjing Qiu [a,b], Caihui Liu [a,b,*], Bowen Lin [c], Xiying Chen [a,b], Duoqian Miao [c]

[a] Department of Mathematics and Computer Science, Gannan Normal University, Ganzhou 34100, Jiangxi, China
[b] Key Laboratory of Data Science and Artificial Intelligence of Jiangxi Education Institutes, Gannan Normal University, Ganzhou 34100, Jiangxi, China
[c] Department of Computer Science and Technology, Tongji University, Shanghai, 201804, China

ARTICLE INFO

Keywords:
Attribute reduction
Adaptive neighborhood rough sets
Three-way decision
Pied kingfisher optimizer

ABSTRACT

Attribute reduction is a critical research topic in rough set theory, aiming to eliminate irrelevant and redundant attributes while maintaining the descriptive power of the data. However, traditional neighborhood rough set-based attribute reduction methods typically require manual setting of the parameters involved in the methods (e.g., neighborhood radius), and often struggle to find the globally optimal feature subset. To address these issues, this paper proposes a novel attribute reduction algorithm (called 3PKO-ANRAR) based on adaptive neighborhood rough sets and the three-way Pied Kingfisher Optimizer (PKO) algorithm. First, the neighborhood relationships are constructed using sample distribution information, and a neighborhood radius reduction factor is defined to enable reasonable adaptation of the neighborhood radius. Second, PKO is introduced as an efficient search mechanism, where the position of the kestrels is treated as the result of attribute reduction (i.e., the reduct). A fitness function is defined based on the attribute dependency of the adaptive neighborhood rough sets to evaluate the quality of the reduct. Third, to mitigate the risk of PKO getting trapped in local optima, several improvements are introduced, including the incorporation of a three-way group partitioning mechanism and a local perturbation strategy, allowing for dynamic updates of the kingfishers' positions during iteration. The proposed attribute reduction algorithm based on adaptive neighborhood rough sets and three-way PKO is able to find feature subsets with minimal information loss while achieving high classification accuracy.

1. Introduction

Attribute reduction is a critical step in data preprocessing. It simplifies models by identifying and removing redundant attributes in datasets (Ibrahim et al., 2020; Wang et al., 2013; Xu & Bu, 2024), thereby enhancing the efficiency and interpretability of algorithms, reducing computational costs, and maintaining the decision-making capability of the datasets (Xu, Guo et al., 2023; Yin et al., 2024). This is crucial for improving the performance of machine learning models and reducing complexity in practical applications (Guo et al., 2024; Liu et al., 2021; Wang et al., 2024).

In the field of data mining and knowledge discovery, the neighborhood rough set model has received significant attention for its unique advantages in handling uncertainty and incomplete data (Liu, Cai et al., 2024; Xia et al., 2024; Yang et al., 2024). With the increasing complexity of data, many scholars have explored various models based on classical rough set theory to handle diverse datasets. Among these, research on attribute reduction using the neighborhood rough set (NRS) method has become increasingly prevalent (Sewwandi et al., 2023; Wang & Zhao, 2024; Xu, Yuan et al., 2023). Yuan et al. proposed a feature selection method that uses a novel zentropy-based uncertainty measure to exploit granular level structures in knowledge space (Yuan et al., 2023). Experimental results demonstrated that their approach achieves state-of-the-art performance in terms of stability and classification accuracy. Wang et al. proposed a feature selection method based on k-nearest neighborhood rough set model to improve the handling of category-mixed samples in heterogeneous data (Wang et al., 2019). Hu et al. introduced a neighborhood rough set model to handle heterogeneous feature subset selection, applying different thresholds for numerical and categorical features (Hu, Yu et al., 2008). Compared to classical techniques, their method showed greater flexibility in dealing with heterogeneous data. Wan et al. (2021) proposed a feature selection method that considers feature interaction in neighborhood rough sets, and developed the Max Relevance minRedundancy MaxInteraction (MRmRMI) evaluation function and the NCMIIFS algorithm.

* Corresponding author at: Department of Mathematics and Computer Science, Gannan Normal University, Ganzhou 34100, Jiangxi, China.
E-mail addresses: wenjingqiu@gnnu.edu.cn (W. Qiu), liucaihui@gnnu.edu.cn (C. Liu), bw_lin@tongji.edu.cn (B. Lin), xiying_chen@gnnu.edu.cn (X. Chen), dqmiao@tongji.edu.cn (D. Miao).

https://doi.org/10.1016/j.eswa.2025.126618
Received 23 November 2024; Received in revised form 7 January 2025; Accepted 19 January 2025
Available online 27 January 2025

Although existing methods have achieved significant success, there are some issues that limit their application (Chen et al., 2019; Tsai et al., 2024). One major issue is the need to manually set a fixed neighborhood radius and apply the same radius to all samples. Many scholars have proposed solutions to address more complex data distribution problems by improving the adaptability of the neighborhood radius. Yuan et al. (2024) introduced a variable precision composite measure within a neighborhood rough set model, adapting neighborhood sizes to enhance computational efficiency and accuracy, by processing uncertain information from both global and local perspectives. Liu, Lin et al. (2024) proposed an adaptive neighborhood rough set model that uses the Sparrow Search Algorithm to optimize the neighborhood radius for each sample, and the search range was defined by the maximum and minimum distances between the target sample and other samples. Yuan, Miao et al. (2024) proposed a zentropy-based uncertainty measure for robust heterogeneous feature selection, leveraging multi-level information integration within heterogeneous neighborhood rough sets. Experimental results confirmed the method's effectiveness and stability compared to existing techniques. Zhou et al. (2019) defined the Gap relation, which is a novel neighborhood rough set approach with adaptive neighbors, and proposed an online streaming feature selection method using the Gap relation in neighborhood rough sets. This method does not require prior domain knowledge or parameter specification. On this basis, Shu et al. (2024) introduced an online hierarchical streaming feature selection algorithm that utilizes adaptive neighborhood rough sets to address dynamic feature spaces. By automatically selecting neighborhood granularity based on hierarchical structures, the method effectively identifies interactive features in high-dimensional data, outperforming existing algorithms in experiments.

However, these methods often overlook the intrinsic distribution characteristics of the data and the impact of the correlation between condition attributes and decision attributes. This indicates the need for a more flexible and adaptive approach to define the neighborhood radius, in order to enhance the model's generalization ability and accuracy.

To address complex optimization problems, the introduction of swarm intelligence algorithms could offer a more robust search mechanism for attribute reduction. The swarm intelligence algorithm typically exhibits outstanding performance and is extensively employed in optimization problems. Chen et al. (2015) proposed a rough set-based feature selection method, which is enhanced by the fish swarm algorithm. This method utilized the search capability of the fish swarm algorithm to identify the optimal feature subset, addressing the limitations of traditional approaches in finding a globally minimal reduct. Tawhid and Ibrahim (2020) proposed a novel binary whale optimization algorithm for feature selection. By simulating the hunting behavior of humpback whales, this algorithm can effectively find the optimal minimum feature subset without relying on the heuristic information. Sun et al. (2023) proposed an adaptive fuzzy neighborhood-based approach to address the oversight of label correlations in multilabel classification, enhancing the prediction efficiency. It also automates the selection of neighborhood radii, reducing manual computation and demonstrating its effectiveness through experiments. Chen et al. (2024) introduced the artificial hummingbird algorithm and three clusters into k-means clustering. Thus, the issues of reducing local optimality and determining the initial clustering center were addressed. Gupta and Gupta (2024) proposed a binary particle swarm optimization method by combining fitness and historical success information and applied it to feature selection. This approach effectively eliminates irrelevant and redundant features while significantly improving the exploration and exploitation capabilities in feature selection.

To sum up, existing neighborhood rough set-based attribute reduction methods face two key limitations: (1) the selection of crucial parameters, such as the neighborhood radius, and (2) the inability to consistently identify globally optimal feature subsets. The neighborhood radius directly influences the balance between granularity and accuracy in data representation. A smaller radius captures finer details but may increase sensitivity to noise, while a larger radius enhances robustness at the cost of losing important distinctions. Thus, appropriately configuring the radius is critical for optimal data analysis. Additionally, identifying the optimal attribute subset in attribute reduction ensures that only the most relevant and informative features are retained. To address these challenges, we propose an attribute reduction algorithm based on three-way PKO and adaptive neighborhood rough sets. This approach overcomes the limitations of fixed neighborhood radii, which are often unsuitable for diverse datasets, and enhances the search for optimal attribute subsets. By leveraging adaptive neighborhood relationships that dynamically adjust to sample distributions, the method improves both granularity and accuracy in data representation. Moreover, the integration of PKO introduces a robust search mechanism inspired by the foraging behavior of pied kingfishers, enabling efficient navigation through complex solution spaces and avoidance of local optima. We further enhance PKO with three-way decision theory, which partitions the population into subgroups to balance exploration and exploitation during the search process. The algorithm is compared with five other neighborhood rough set-based methods in terms of runtime and attribute subset quality, validated using Decision Tree (DT) and K-nearest Neighbors (KNN) classifiers. Experimental results show that 3PKO-ANRAR consistently selects optimal attribute subsets with relatively short runtimes. The main contributions of this study are summarized as follows:

(1) We apply the distribution radius to address the challenge of setting the neighborhood radius across different types of datasets in classical neighborhood rough sets, and construct a new neighborhood relationship. By leveraging sample distribution information, we define a neighborhood radius reduction factor to adaptively adjust the neighborhood radius.

(2) We apply PKO to the search of attribute subsets. The positions of the kestrels are treated as the reducts, and a fitness function is constructed based on the attribute dependency of the adaptive neighborhood rough set to evaluate the quality of the reducts, aiming to identify the optimal reduct.

(3) We propose a population-based three-way partitioning mechanism and apply a local perturbation strategy to prevent the PKO algorithm from becoming trapped into local optima. These enhancements dynamically update the search strategy during iterations, thereby improving the robustness of the algorithm.

(4) We propose an attribute reduction algorithm (called 3PKO-ANRAR) based on the adaptive neighborhood rough set and three-way PKO algorithm. By comparing 3PKO-ANRAR with other methods, the final selected attribute subsets of the proposed algorithm exhibit higher accuracy and balance precision. We demonstrate the proposed algorithm's effectiveness, particularly its relatively lower runtime and superior robustness in complex environments.

The organization of the sections in this paper is delineated as follows. The neighborhood rough sets and the PKO algorithm are reviewed in Section 2. Section 3 provides a detailed explanation of the proposed algorithm, including the improved PKO algorithm and the adaptive neighborhood rough set model. In Section 4, a comparative experiment is conducted to evaluate the efficacy of the proposed algorithm. Finally, Section 5 concludes the paper and offers insights for future research directions.

2. Preliminaries

This section provides crucial background knowledge and two key approaches, the neighborhood rough set model and the pied kingfisher optimizer.

2.1. Neighborhood rough set

In rough set theory, an information table is formally represented as a 2-tuple $\langle U, A\rangle$, where $U = \{\kappa_1, \kappa_2, \ldots, \kappa_n\}$ denotes the set of samples, also referred to as the universe, and the set A comprises the attributes of these samples. If A is partitioned into C and D, where C represents the set of condition attributes and D represents the set of decision attributes, then the information table $\langle U, A\rangle$ is called a decision table, denoted as $DS = (U, C \cup D)$.

Let $DS = (U, C \cup D)$ be a decision system. For each $B \subseteq C$ and $\kappa_i, \kappa_j \in U$, let δ be the neighborhood radius of κ_i, and let $Edis_B(\kappa_i, \kappa_j)$ represent the Euclidean distance between κ_i and κ_j under B. The neighborhood of κ_i with respect to B is defined as follows:

$$\delta_B(\kappa_i) = \{\kappa_j \in U | Edis_B(\kappa_i, \kappa_j) \leq \delta\}. \tag{1}$$

Let $DS = (U, C \cup D)$ be a decision system. For each $B \subseteq C$, $\kappa_i \in U$ and $X \subseteq U$, the neighborhood upper and lower approximations of X under B can be defined as follows:

$$\overline{N_B}(X) = \{\kappa_i \in U | \delta_B(\kappa_i) \cap X \neq \varnothing\}, \tag{2}$$

$$\underline{N_B}(X) = \{\kappa_i \in U | \delta_B(\kappa_i) \subseteq X\}, \tag{3}$$

where the lower approximation of X under B is commonly referred to as the positive region of X under B, denoted as $POS_B(X)$. $POS_B(X)$ represents the subset of elements in X that can be definitively classified as members of X based on the information provided by B.

Let $DS = (U, C \cup D)$ be a decision system. The neighborhood conditional entropy reflects the uncertainty of attribute set B for decision attribute D. For each $B \subseteq C$, The neighborhood conditional entropy of D on B is defined as follows:

$$NE(D|B) = NE(D \cup B) - NE(B), \tag{4}$$

where $NE(B)$ is the neighborhood information entropy of B on U, $NE(B) = 1 - \frac{1}{|U|}\sum_{i=1}^{|U|}\frac{|N_B^\delta(\kappa_i)|}{|U|}$, $N_B^\delta(\kappa_i)$ is a neighborhood of sample κ_i with respect to B.

2.2. Pied kingfisher optimizer

The Pied Kingfisher Optimizer (PKO) algorithm represents a novel bio-inspired optimization strategy, drawing its design inspiration from the distinctive flight patterns and intelligent foraging behaviors of the pied kingfisher. Based on a meticulous examination of the pied kingfisher's predatory behaviors, this algorithm delineates three pivotal stages: perching and hovering strategies (exploration stage), diving strategy (exploitation stage), and commensalism stage (local escape stage). According to the comparative analysis presented in Bouaouda et al. (2024), the PKO algorithm adeptly navigates solution spaces in intricate optimization quandaries, steadily converging towards the optimal solution while upholding diversity.

2.2.1. Initialization

In alignment with the majority of population-based swarm intelligence algorithms, Pied kingfisher optimizer initiates the search process by generating random initial solutions within the defined search space. This approach serves as the preliminary step in exploring potential solutions. The generation of the initial population can be defined as follows:

$$h_{i,j} = Low + (Up - Low) \times rand, \tag{5}$$

where $1 \leq i \leq N$ and $1 \leq j \leq$ Dim (Note: N represents the population size, and Dim indicates the problem's dimensionality). $h_{i,j}$ signifies the position of the ith individual in the jth dimension, and $rand$ is a stochastic value in the range of $[0, 1]$, Up denote the upper limits of the search space, and Low denote the lower limits of the search space.

2.2.2. Perching and hovering strategies (exploration stage)

The exploration phase of the PKO algorithm is inspired by the perching and hovering behaviors of the pied kingfisher. Observations in their natural habitats reveal that these birds alternate between striking from perches and attacking while hovering. The PKO algorithm dynamically modifies the positions of search agents in accordance with the foraging behaviors of pied kingfishers. This adjustment is formally defined as follows:

$$h_i(z+1) = h_i(z) + \alpha \times T \times (h_j(z) - h_i(z)), \tag{6}$$

where $1 \leq i \neq j \leq N$ (Note: N denotes the population size), $h_i(z+1)$ denotes the position of the ith agent in the subsequent iteration, and $h_i(z)$ denotes its current position.

In Eq. (6), the parameter α is defined as: $2 \times randn\ (1, Dim) - 1$, where $randn$ is a normally distributed random variable and Dim denotes the problem's dimensionality. Moreover, the value of parameter T is dynamically adjusted according to the current strategy, which can be either 'Perching' or 'Hovering'.

The PKO algorithm models the bird's behavior of perching on natural and artificial structures to precisely capture prey. In the perching strategy, the parameter T is computed as follows:

$$T = (\exp(1) - \exp(\frac{z-1}{Max\_Iter})^{\frac{1}{BF}}) \times \cos(Crest\_angles), \tag{7}$$

where $Crest\_angles = 2 \cdot pi \cdot rand$, BF (Beating Factor) is a constant with a value of 8, and $rand$ is a uniformly distributed random variable between 0 and 1.

On the contrary, during the hovering phase, the Pied Kingfisher maintains stability through rapid wing flapping. In this hovering strategy, the parameter T is determined as follows:

$$T = beating\_rate \times (\frac{t^{\frac{1}{BF}}}{Max\_Iter^{\frac{1}{BF}}}), \tag{8}$$

where $beating\_rate = rand \times \frac{fit_{PKO}(j)}{fit_{PKO}(i)}$, BF is set to 8, $rand$ is a random value between 0 and 1, and $fit_{PKO}(i)$ denotes the fitness of the ith individual within the population.

2.2.3. Diving strategy (exploitation stage)

The exploration phase of the PKO algorithm is inspired by the perching and hovering movements of the pied kingfisher. Observations in their natural settings reveal that these birds alternate between striking from perches and launching attacks from hovering postures. The PKO algorithm dynamically modifies the placements of search agents in accordance with the foraging activities of pied kingfishers. This modification is officially delineated as follows:

$$h_i(z+1) = h_i(z) + HA \times \sigma \times \alpha \times (b - h_{best}(z)), \tag{9}$$

where $1 \leq i \leq N$, α is a control parameter which is defined as: $\alpha = 2 \times randn\ (1, Dim) - 1$, $\sigma = \exp(\frac{-t}{Max\_Iter})^2$, and $b = h_i(z) + \sigma^2 \times randn \cdot h_{best}(z)$.

In Eq. (9), HA denotes the hunting ability which is defined as: $HA = rand \cdot (\frac{fit_{PKO}(i)}{Best\_fit_{PKO}})$, where $fit_{PKO}(i)$ denotes the fitness of the ith individual within the population, and $Best\_fit_{PKO}$ represents the highest fitness value achieved across all iterations.

2.2.4. Commensalism stage (local escape stage)

The pied kingfisher exhibits a symbiotic interaction with several otter species. This relationship indicates mutual benefit between both species without inflicting harm. This symbiotic behavior can be formally depicted as follows:

$$h_i(z+1) = \begin{cases} h_m(z) + \sigma \times \alpha \times abs(h_i(z) - h_n(z)), & if\ rand > (1 - PE), \\ h_i(z), & otherwise, \end{cases} \tag{10}$$

where $1 \leq i \leq N$, α is a control parameter which is defined as: $\alpha = 2 \times randn\ (1, Dim) - 1$, $\sigma = \exp(\frac{-z}{Max\_Iter})^2$, and $h_m(z)$ and $h_n(z)$

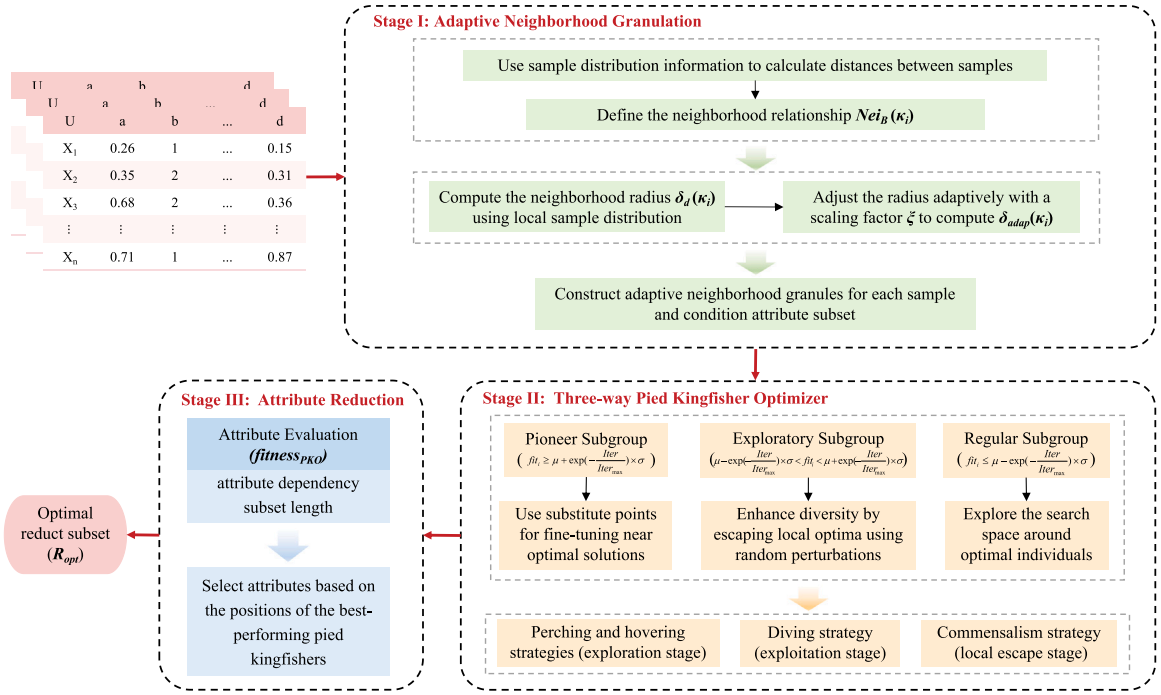


Fig. 1. The framework of this 3PKO-AMRAR model.

represent the positions of two individuals randomly selected from the population.

In Eq. (10), PE denotes the predatory efficiency of the pied kingfisher, which is defined as: $PE = PE_{\max} - (PE_{\max} - PE_{\min}) \times (\frac{z}{Max\_Iter})$, where the constant $PE_{\max}$ is set to 0.5 and the constant $PE_{\min}$ is set to 0.

3. The proposed 3PKO-ANRAR model

To overcome the problem of setting the neighborhood radius across diverse datasets in rough set theory, we introduce a distribution radius and define a neighborhood radius adjustment factor to calculate the optimal neighborhood radius for each sample with the dataset. This optimal radius is used to construct an adaptive neighborhood rough set, applied to attribute reduction. Next, the PKO algorithm is introduced as a search mechanism for attribute subsets. The positions of the pied kingfisher are treated as the reducts, and a corresponding fitness function is defined to evaluate the quality of the reducts. Additionally, to prevent the PKO algorithm from becoming trapped into local optima, the concept of three-way decision is incorporated into the PKO's population division mechanism. The population is divided in three ways, with strategies dynamically updated over multiple iterations, aiming to discover the globally optimal reduct. This work develops an attribute reduction algorithm based on adaptive neighborhood rough sets and three-way pied kingfisher optimizer, and illustrates the framework in Fig. 1 from three phases: (1)adaptive neighborhood granulation; (2) three-way Pied Kingfisher Optimizer; (3) attribute reduction.

3.1. Adaptive neighborhood radius

In various datasets, the distribution of samples is not uniform, making it challenging to set a consistent radius for selection. Consequently, Zhou et al. (2019) introduced a novel neighborhood relationship: the neighborhood is automatically determined based on the distribution of samples surrounding the target sample.

Definition 1. Given a decision system $DS = (U, C \cup D)$, for each $B \subseteq C$ and $\kappa_i \in U$, let $Nei_B(\kappa_i)$ be a set of neighbors of κ_i arranged in ascending order of their distances from κ_i under B, which can be represented as follows:

$$Nei_B(\kappa_i) = \{\kappa_i^1, \kappa_i^2, \ldots, \kappa_i^k, \ldots, \kappa_i^{n-1}\}, \tag{11}$$

where $Edis_B(\kappa_i, \kappa_i^1) \le Edis_B(\kappa_i, \kappa_i^2) \le \cdots \le Edis_B(\kappa_i, \kappa_i^{n-1})$, and $Edis_B$ denotes the distance metric under B.

The samples in the set $Nei_B(\kappa_i)$ are arranged in ascending order of their distances from κ_i under B, as illustrated in Fig. 2.

From Fig. 2, it can be observed that there are $n-2$ intervals between adjacent samples in the set $Nei_B(\kappa_i)$, where the first interval is defined as: $H_1 = Edis_B(\kappa_i, \kappa_i^2) - Edis_B(\kappa_i, \kappa_i^1)$. Moreover, the second interval is defined as: $H_2 = Edis_B(\kappa_i, \kappa_i^3) - Edis_B(\kappa_i, \kappa_i^2)$, and so on. Consequently, the mth interval is defined as: $H_m = Edis_B(\kappa_i, \kappa_i^{m+1}) - Edis_B(\kappa_i, \kappa_i^m)$.

Assuming an even distribution from κ_i^1 to κ_i^{n-1}, we partition the interval $Edis_B(\kappa_i, \kappa_i^{n-1})$ into $n-1$ equal segments: $H_1, H_2, \ldots, H_{n-1}$, such that $Width_h = Width_2 = \cdots Width_{n-1} = h$. Each segment h contains exactly one sample. It is acknowledged that the distribution from κ_i^1 to κ_i^{n-1} is inherently non-uniform. Between κ_i^1 and κ_i^{n-1}, if the distance between any two instances κ_i^k and κ_i^{k+1} exceeds h, this is termed a "h_{GAP}", denoted as $h_{GAP}(\kappa_k, \kappa_{k+1})$. Consequently, the samples between κ_i and the first encountered h_{GAP} are considered the nearest neighbors of κ_i. In Zhou et al. (2019), $h_{GAP} = 1.5 \times \frac{Edis_{\max} - Edis_{\min}}{n-1}$.

Definition 2. Given a decision system $DS = (U, C \cup D)$, for each $B \subseteq C$ and $\kappa_i \in U$, the distance from sample κ_i to the location of the first gap is defined as the neighborhood radius, which can be represented as follows:

$$\delta^d(\kappa_i) = Edis_B(\kappa_i, \kappa_i^m), \tag{12}$$

where $Edis(\kappa_i, \kappa_i^m) > h_{GAP}$ and for any $2 \le k \le m$, $Edis_B(\kappa_i^k, \kappa_i^{k+1}) < h_{GAP}$.

From Fig. 2, it is known that $H_m = Edis_B(\kappa_i, \kappa_i^{m+1}) - Edis_B(\kappa_i, \kappa_i^m) > h_{GAP}$ and for any $1 < k < m-1$, $H_k < h_{GAP}$. Next, we will illustrate the computation process of the distribution radius through an example.

The distribution radius is calculated based on the local distribution of samples, ignoring the inter-class distribution within the sample space. If there is an overly dense region of samples, the chosen distribution radius may be too large, resulting in an excessive number

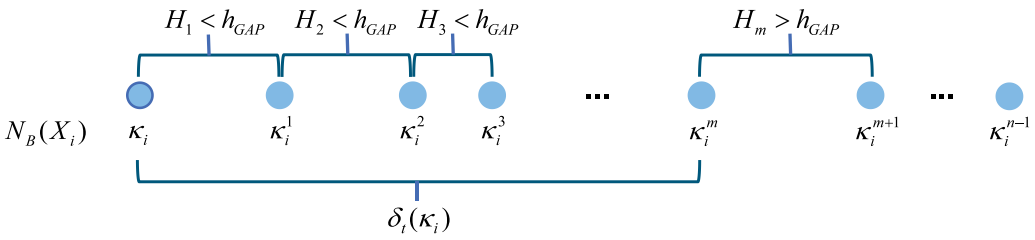


Fig. 2. Contrast difference between the target and background.

of samples within the neighborhood. This not only increases the computation time for the neighborhood radius, but also leads to overly coarse granulation. To address this issue, considering the global sample distribution, a maximum value is set for the distribution radius in this paper.

Definition 3. Given a decision system $DS = (U, C \cup D)$. For each $B \subseteq C$ and $\kappa_i \in U$, consider the distribution of all sample labels and distributions in the sample space. The maximum value of the distribution radius is defined as follows:

$$\delta^d_{\max}(\kappa_i) = \frac{1}{2}\left|\mathbb{C}_{SameLabel}(\kappa_i, B) - \mathbb{C}_{DiffLabel}(\kappa_i, B)\right|, \tag{13}$$

where $\mathbb{C}_{SameLabel}(\kappa_i, B)$ represents the center of samples with the same label as κ_i under B, and $\mathbb{C}_{DiffLabel}(\kappa_i, B)$ represents the center of samples with different labels from κ_i under B.

$\mathbb{C}_{SameLabel}(\kappa_i, B)$ and $\mathbb{C}_{DiffLabel}(\kappa_i, B)$ are defined as follows:

$$\mathbb{C}_{SameLabel}(\kappa_i, B) = \frac{\sum_{i=1}^{|P_{same}|} w_i g(\kappa_i, B)}{\sum_{i=1}^{|P_{same}|} w_i}, \tag{14}$$

$$\mathbb{C}_{DiffLabel}(\kappa_i, B) = \frac{\sum_{i=1}^{|P_{diff}|} w_i g(\kappa_i, B)}{\sum_{i=1}^{|P_{diff}|} w_i}, \tag{15}$$

where P_{same} denotes the set of samples with the same label as κ_i, and P_{diff} denotes the set of samples with different labels from κ_i. The weight $w_i = \frac{1}{Edis_B(\kappa_i, \kappa_j)}$ considers the distribution between classes in the sample space. Specifically, the maximum value of the distribution radius should not exceed the distance between the center of its class and other class centers; otherwise, it is considered to exceed the class's distribution range.

Compared to the traditional granulation mechanism with fixed radii in neighborhood rough sets, this approach allows for the adaptive determination of a neighborhood radius for each sample based on its distribution. Suppose that $Class_1$ and $Class_2$ are considered the most suitable for classification and are expected to be selected during attribute reduction. In that case, it is necessary to ensure that the number of samples correctly classified under these two attributes is higher than other attributes. However, as shown in Fig. 3, the neighborhoods of the samples w_i and v_i at the boundary between two classes may still contain samples from different classes. In such case, w_i and v_i do not belong to the identified samples and cannot be correctly classified. If there are many such samples under $Class_1$ and $Class_2$, the importance of these attributes may be underestimated during evaluation, potentially leading to their exclusion.

Definition 4. Given a decision system $DS = (U, C \cup D)$. For each $B \subseteq C$ and $\kappa_i \in U$, the adaptive neighborhood radius adjustment factor $\xi(\kappa_i)$ for sample κ_i is defined as follows:

$$\xi(\kappa_i) = \frac{1}{1 + NE(D \mid B)}, \tag{16}$$

where $\xi(\kappa_i) > 0$, $NE(D \mid B)$ represents the neighborhood conditional entropy, which quantifies the uncertainty of the decision attribute D with respect to the condition attribute subset B. The neighborhood

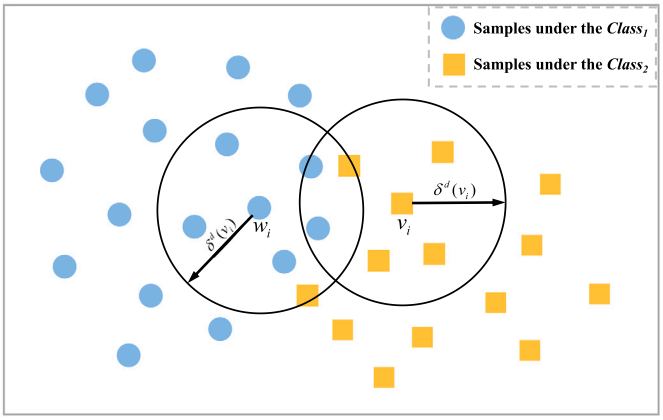


Fig. 3. Boundary issues in sample classification.

conditional entropy is given by $NE(D \mid B) = NE(D \cup B) - NE(B)$, where $NE(B)$ is the neighborhood information entropy of B on U. Specifically, $NE(B) = 1 - \frac{1}{|U|}\sum_{i=1}^{|U|}\frac{|N_B^\delta(\kappa_i)|}{|U|}$, where $N_B^\delta(\kappa_i)$ denotes the neighborhood of sample κ_i with respect to B.

Definition 5. Given a decision system $DS = (U, C \cup D)$, For each $\kappa_i \in U$, the adaptive neighborhood radius of κ_i is defined as follows:

$$\delta^d_{adap}(\kappa_i) = \xi \times \delta^d(\kappa_i), \tag{17}$$

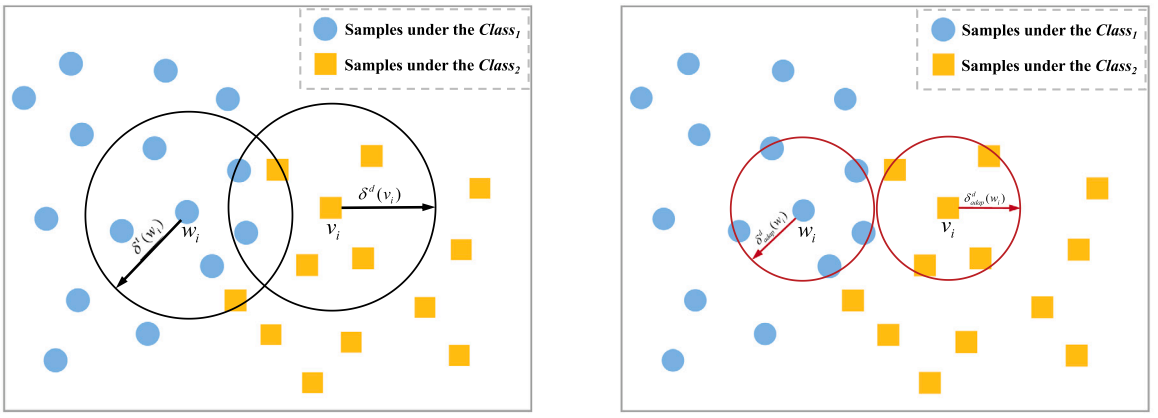
where ξ is the adaptive neighborhood radius adjustment factor.

During the neighborhood adjustment process, if the adjustment ratio is too large, the neighborhood radius becomes too small, resulting in the absence of other samples within the neighborhood. On the contrary, if the adjustment ratio is too small, too many heterogeneous samples remain in the neighborhood, resulting in ineffective adjustment. Therefore, the range of $\delta^d_{adap}(\kappa_i)$ is as follows:

$$\begin{cases} \delta^d_{adap} = 0.1\delta^d, & \xi < 0.1; \\ \delta^d_{adap} = \xi \times \delta^d, & 0.1 \le \xi \le 0.9; \\ \delta^d_{adap} = 0.9\delta^d, & \xi > 0.9. \end{cases} \tag{18}$$

When $\xi < 0.1$, it is considered that the maximum adjustment strength has been reached, and the parameter $\xi = 0.1$ is used. When $\xi > 0.9$, the condition attribute subset has little correlation with the decision attribute, so the adjustment strength is reduced, and the parameter $\xi = 0.9$ is applied. When $0.1 \le \xi \le 0.9$, the neighborhood radius is reduced according to the value of ξ given by the formula.

Fig. 4 illustrates the granulation effects on boundary samples before and after adjusting the neighborhood radius. In Fig. 4(a), a fixed neighborhood radius δ^d is applied to the samples w_i and v_i, encompassing samples from both $Class_1$ and $Class_2$ within its neighborhood. This overlap introduces ambiguity in the classification of w_i and v_i, making it difficult to determine its correct classes. The fixed radius fails to adapt to the local sample distribution, resulting in coarse granularity



(a) Boundary issues in sample classification (b) Adaptive neighborhood radius adjustment

Fig. 4. Adaptive neighborhood radius adjustment and sample granulation effects.

and potential misclassification at class boundaries. To address this issue, an adaptive neighborhood adjustment strategy is adopted. Since condition attributes have varying correlations with decision attributes, the neighborhood radius is adjusted based on these correlations. Specifically, a higher correlation results in a greater adjustment to enhance the precision of granulation, whereas a lower correlation reduces the adjustment. This approach helps in selecting more suitable attributes for classification. As shown in Fig. 4(b), the adaptive neighborhood radius dynamically adjusts based on local density and the dependency between condition and decision attributes. This adaptation reduces boundary ambiguity and ensures that w_i is correctly classified as belonging to $Class_1$. Similarly, for v_i, the adaptive radius δ^d_{adap} excludes blue samples from its neighborhood, ensuring its accurate classification as belonging to $Class_2$. By fine-tuning the radius according to local density and class distribution, the adaptive mechanism achieves finer granularity and improved classification accuracy.

3.2. Adaptive neighborhood rough set

Definition 6. Given a decision system $DS = (U, C \cup D)$, for each $B \subseteq C$ and $\kappa_i, \kappa_j \in U$, let δ^D_{adap} be the adaptive neighborhood radius of κ_i. Then, the adaptive neighborhood of κ_i with respect to conditional attribute subset B is defined as follows:

$$\delta^D_{adap}(\kappa_i) = \{\kappa_j \in U | Edis_B(\kappa_i, \kappa_j) \leq \delta^D_{adap}\}, \tag{19}$$

where $Edis_B(\kappa_i, \kappa_j)$ is the Euclidean distance between κ_i and κ_j regarding attribute subset B.

Next, it is necessary to develop two approximation operators for the purpose of constructing an approximate representation of the target samples' neighborhoods.

Definition 7. Given a decision system $DS = (U, C \cup D)$, for each $B \subseteq C$ and $\kappa_i, \kappa_j \in U$. Let $U/D = \{D_1, D_2, \ldots, D_k\}$, the upper and lower approximations of D with respect to conditional attribute subset B can be defined as follows:

$$\overline{AdapN_B}(D) = \bigcup_{i=1}^{k} \overline{AdapN_B}(D_i), \tag{20}$$

$$\underline{AdapN_B}(D) = \bigcup_{i=1}^{k} \underline{AdapN_B}(D_i). \tag{21}$$

for any $1 \leq i \leq k$,

$$\overline{AdapN_B}(D_i) = \{\kappa \in U | \delta^D_{adap}(\kappa) \cap D_i \neq \varnothing\}, \tag{22}$$

$$\underline{AdapN_B}(D_i) = \{\kappa \in U | \delta^D_{adap}(\kappa) \subseteq D_i\}. \tag{23}$$

Definition 8. Given a decision system $DS = (U, C \cup D)$, for each $B \subseteq C$ and $\kappa_i, \kappa_j \in U$. The positive, boundary, and negative regions of D with respect to B can be defined as follows:

$$POS_{AdapN_B}(D) = \underline{AdapN_B}(D), \tag{24}$$

$$BND_{AdapN_B}(D) = \overline{AdapN_B}(D) - \underline{AdapN_B}(D), \tag{25}$$

$$NEG_{AdapN_B}(D) = U - \overline{AdapN_B}(D). \tag{26}$$

Definition 9. Given a decision system $DS = (U, C \cup D)$, for each $B \subseteq C$, the dependency degree of D with respect to B is defined as follows:

$$ADE(D, U) = \frac{\left|POS_{AdapN_B}(D)\right|}{|U|}. \tag{27}$$

It is obvious that $0 \leq ADE(D, U) \leq 1$, and the dependency degree calculates the proportion of samples that can be accurately classified under conditional attribute subset B.

3.3. Three-way pied kingfisher optimizer

In this subsection, we introduce the integration of three-way decision (Yao, 2011) into the PKO algorithm. The three-way decision is crucial for dynamically categorizing the kingfisher population into pioneer, regular, and exploratory subgroups. This categorization is based on the fitness of individuals relative to the population's mean and standard deviation. By incorporating these decision-making strategies, the algorithm effectively balances exploration and exploitation. The pioneer subgroup focuses on fine-tuning solutions near optimality, the regular subgroup facilitates diverse exploration, and the exploratory subgroup enhances overall diversity. This integration not only improves convergence speed but also mitigates the risk of local optima.

Definition 10. Let $X_i = \{\kappa_{i,1}, \kappa_{i,2}, \ldots, \kappa_{i,d}\}$ be the position of a pied kingfisher. Through a specific transformation method, the continuous position X_i can be converted into a binary position of length d, denoted as $X'_i = \{\kappa'_{i,1}, \kappa'_{i,2}, \ldots, \kappa'_{i,d}\}$, where d represents the number of attributes. This specific transformation method can be defined as follows:

$$\kappa'_{i,j} = \begin{cases} 1 & if\ rand < \kappa_{i,j} \\ 0 & otherwise \end{cases} \tag{28}$$

where *rand* is a random number between 0 and 1, $\kappa_{i,j} \in [0,1]$ denotes the jth dimension value of prey i in the continuous space. Each $\kappa'_{i,j}$ corresponds to an attribute, with a value of 1 indicating that the attribute is selected, and a value of 0 indicating that it is not selected.

Definition 11. The fitness function is defined based on attribute dependency and the length of the reduced set. This function evaluates the quality of feature subsets, with larger fitness values indicating superior feature subsets. The fitness function can be defined as follows:

$$fit_{PKO} = \alpha \times ADE_R(D) + \beta \times RLEN, \tag{29}$$

where $RLEN$ is the length of the reduced set, calculated as $RLEN = \frac{|C|-|R|}{|C|}$, $|R|$ is the length of the currently selected attribute subset, and $|C|$ is the total number of attributes in the dataset.

In Eq. (27), the parameters α and β correspond to the importance of $ADE_R(D)$ and $RLEN$, respectively, where $\alpha \in [0,1]$ and $\beta = 1 - \alpha$. Attribute dependency, representing the proportion of the positive region, is more crucial than the length of the reduced set to ensure classification accuracy, as it allows the derivation of deterministic rules.

Definition 12. By incorporating the concept of three-way decision-making, individuals in the kingfisher population are dynamically divided into pioneer, regular, and exploratory subgroups based on the relationship between individual fitness and the mean and standard deviation of the population fitness. Individuals with fitness above the threshold are classified into the pioneer subgroup, those below the threshold are classified into the regular subgroup, and the remaining individuals are classified into the exploratory subgroup. The partitioning rules are as follows.

$$\begin{cases} fit_i \geq \mu + \exp(-\frac{Iter}{Iter_{\max}}) \times \sigma, & i \rightarrow POP_p(t), \\ \mu - \exp(-\frac{Iter}{Iter_{\max}}) \times \sigma < fit_i < \mu + \exp(-\frac{Iter}{Iter_{\max}}) \times \sigma, & i \rightarrow POP_e(t), \\ fit_i \leq \mu - \exp(-\frac{Iter}{Iter_{\max}}) \times \sigma, & i \rightarrow POP_r(t), \end{cases} \tag{30}$$

where $Iter$ is the current iteration number, $Iter_{\max}$ is the maximum iteration number, fit_i denotes the fitness value of the ith individual, μ is the mean fitness of the current population, and σ is the standard deviation of the population fitness. As the algorithm iterates, $\exp(-\frac{Iter}{Iter_{\max}})$ gradually decreases, tightening the criteria for individual partitioning. Initially, the criteria are more lenient, allowing more individuals to enter the pioneer subgroup, while later, only those individuals who are very close to the optimal individuals enter the pioneer subgroup.

Definition 13. For individuals in the regular subgroup $POP_r(t)$, the kingfishers are relatively far from the current optimal individual. As they approach the optimal individual, there are numerous possibilities that make them less prone to local optima. In contrast, kingfishers in the pioneer subgroup $POP_p(t)$ are closer to the optimal individual and may quickly converge to a local optimum. Thus, by constructing substitute points for the optimal individual, the position update process of kingfishers in the pioneer subgroup utilizes the substitute point X_m instead of the optimal individual, which is defined as follows:

$$X_m = X - m \times c_i \times \left| X - X_{rand} \right|, \tag{31}$$

where $m = \frac{Iter_{\max} - Iter}{Iter_{\max}}$, and c_i is the perturbation amplitude which is dynamically adjusted based on the relative difference in individual fitness. Individuals with a larger fitness gap will have greater perturbation, while those with a smaller gap will have less perturbation, defined as: $c_i = \frac{fit_{best} - fit_i}{fit_{best}}$. Moreover, X is the current optimal individual, and X_{rand} is a randomly generated individual. For kingfishers in the pioneer subgroup, the position update process uses X_m to replace the optimal individual, while the position update process for individuals in the regular subgroup remains unchanged.

3.4. The 3PKO-ANRAR algorithm

In this subsection, we propose an attribute reduction algorithm based on the adaptive neighborhoods and three-way pied kingfisher optimizer(denoted by 3PKO-ANRAR).

To illustrate the procedure of algorithm, the neighborhood distribution radius and the radius adjustment factor are computed first, so as to obtain each sample's adaptive neighborhood radius. Then the neighborhood conditional entropy and neighborhood dependence between attribute and label sets are calculated. The PKO algorithm is then introduced to attribute reduction. Considering that in PKO, the individual's move decision is probabilistic, that is, different search strategies are used to update the location under different probabilities. Therefore, this uncertainty is eliminated by combining three-way decisions. Finally, an attribute reduction algorithm based on the improved three-way PKO is developed. After several iterations, the optimal reduction is selected and presented in Algorithm 1.

The algorithm 1 is briefly discussed as follows. Step 1 randomly generates a set of solutions, and steps 2–4 calculate the adaptive neighborhood radius for each sample. Steps 5–24 are used to get the final reduction. Among them, steps 5–9 are to calculate the fitness function of the current sample and get the current elite individual. Steps 10–23 are the search scheme of the three-way PKO algorithm. The algorithm stops until the maximum number of iterations is reached. Steps 21–22 update the current location only if the new location offers better solution than the current location. Since the Euclidean distance employed in this paper is not suitable for measuring distances between high-dimensional vectors, high-dimensional datasets are excluded from consideration. Fig. 5 clearly shows the flow diagram of the algorithm.

In the adaptive neighborhood radius computation stage, calculating pairwise distances for a dataset with samples and attributes requires $O(|U|^2 \times |C|)$. Sorting the distances for each sample to determine the neighborhood radius is $O(|U|^2 \log |U|)$. Adjusting the radius with scaling factors involves a linear pass over the samples, contributing $O(|U|)$. Thus, this stage has a total complexity of $O(|U|^2 \times |C|) + O(|U|^2 \times \log |U|)$.

In the fitness function evaluation stage, computing dependency measures like the positive region involves examining the neighborhood relationships of all samples, contributing $O(|U|^2 \times |C|)$. Constructing the fitness function for a population of size N adds $O(N \times |C|)$. Therefore, the total complexity of this stage is $O(|U|^2 \times |C|) + O(N \times |C|)$.

The three-way PKO optimization stage includes initializing the population, dividing it into subgroups, and updating positions over Itermax iterations. Initialization and position updates each take $O(N \times |C|)$, and sorting for subgroup division takes $O(N \log N)$. Repeating these steps for Itermax iterations results in a total complexity of $O(Iter_{\max} \times N \times |C|)$ (since $N \log N$ is dominated by $N \times |C|$ when $|C|$ is large).

The attribute reduction and output stage requires scanning the final solution to extract the optimal subset, with a complexity of $O(|C|)$. Combining these, the overall time complexity of the 3PKO-ANRAR algorithm is $O(|U|^2 \times |C|) + O(|U|^2 \times \log |U|) + O(Iter_{\max} \times N \times |C|)$. Assuming $|U|$ and $|C|$ are proportional to the problem size, and Itermax and N are constants or grow more slowly compared to the problem size, the terms $|U|^2 \times |C|$ and $|U|^2 \log |U|$ will be more significant than Itermax $\times N \times |C|$. Since $\log |U|$ grows slower than $|C|$, as $|U|$ becomes very large, the term $|U|^2 \times |C|$ will dominate. Therefore, the time complexity of the given expression is $O(|U|^2 \times |C|)$.

4. Experimental design and analysis

This section evaluates the effectiveness of 3PKO-ANRAR by comparing it with five representative algorithms. The comparative analysis is divided into four parts: classification performance, running time, the number of attributes selected, and statistical tests.

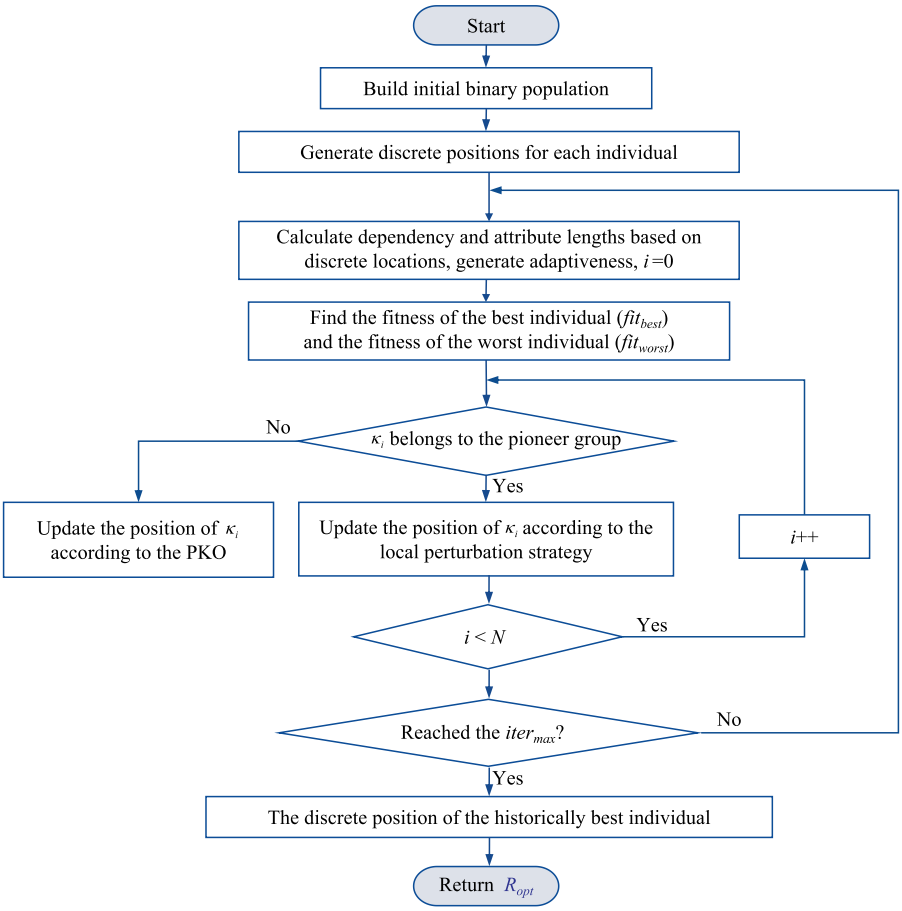


Fig. 5. The flow chart of Algorithm 1.

4.1. Experimental environment and datasets

All experiments were performed on MATLAB2023b with the following configuration. Windows 10 operating system, with the processor being Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz, and 16.0 GB of RAM. To verify the effectiveness of 3PKO-ANRAR, 16 datasets were chosen from the UCI Machine Learning Repository[1] and gene datasets[2] for comparative experiments. Detailed information on the experimental datasets is shown in Table 1.

4.2. Experimental scheme

This subsection introduces the experimental setup. We compared the proposed algorithm with raw datasets and five existing algorithms on 16 datasets listed in Table 1. In the PKO algorithm, the number of iterations was set to 20. and all the conditional attributes within the data set have been normalized to a range of [0, 1].

To validate the effectiveness of 3PKO-ANRAR algorithm, the experiments assessed the algorithm from three perspectives:

(1) evaluating the attribute reduction process;

(2) evaluating whether 3PKO-ANRAR algorithm can effectively remove redundant attributes while improving the accuracy of raw classification;

(3) evaluating whether the efficiency of 3PKO-ANRAR algorithm outperforms the five representative attribute reduction algorithms, i.e., NNRS (Wang et al., 2019), HARNRS (Hu, Yu et al., 2008), Far-VPKNN (Hu, Liu et al., 2008), HARCD (Wang et al., 2018), and FSMRI (Qu et al., 2023).

For (1), the evaluation of the reduction process was conducted by comparing the runtime of each algorithm. For (2) and (3), the accuracy and balanced accuracy of the attribute groups selected by each algorithm were assessed using Decision Tree (DT) and K-Nearest Neighbors (KNN) classifiers, with 10-fold cross-validation applied to each dataset. The Gini index was employed as the partition criterion for the Decision Tree, and the parameter K in KNN was varied from 1 to 10.

Accuracy(ACC) is a fundamental metric for assessing classification performance. It is defined as the proportion of correctly classified instances out of the total number of instances evaluated. It can be calculated according to Eq. (32).

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{32}$$

where TP refers to instances where positive samples are accurately identified as positive, FN denotes cases where positive samples are mistakenly classified as negative, FP describes scenarios where negative samples are incorrectly identified as positive, and TN signifies instances where negative samples are correctly classified as negative.

Accuracy measures the overall correctness of predictions and is widely utilized for its simplicity and interpretability. However, it should be supplemented with additional metrics, such as balanced accuracy, to offer a more comprehensive evaluation of a model's performance, particularly in scenarios with data imbalance. Balanced accuracy(BA) provides a thorough assessment by considering sensitivity across all classes. It is calculated as shown in Eq. (33).

$$\mathrm{BA} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}\right), \tag{33}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. This metric accounts for

[1] http://archive.ics.uci.edu/ml/datasets.php.

[2] https://csse.szu.edu.cn/staff/zhuzx/datasets.html.

Algorithm 1: 3PKO-ANRAR

Input: Decision table $DS = (U, C \cup D)$, $Iter_{max}$
Output: *An optimal reduct* R_{opt}

Initialization of population creation X_i $(1 \le i \le N)$
for *any sample* κ_i *in* U **do**
  Generate the adaptive neighborhood radius δ_{adap} of κ_i according to Equation (19).
end
Generate the current pied kingfisher group $POP(iter) = X_1^{iter}, X_2^{iter}, \ldots, X_N^{iter}$.
for $iter = 1, 2, \ldots, Iter_{max}$ **do**
  Use Equation (29) to calculate the fitness of each individual in $POP(iter)$.
  Search for the position of the current optimal individual X_{best} and record its fitness value fit_{best}.
  Use Equation (30) to calculate the probability of the optimal replacement group.
  for $i = 1 : N$ **do**
    Calculate the adaptive distance from individual X_i to the optimal individual R_{opt}.
    if $rand() < 0.8$ **then**
      Update the position of pied kingfisher X_i according to Equation (6).
    else if $rand() > 1 - PE$ **then**
      Update the position of pied kingfisher X_i according to Equation (9).
    else
      Update the position of pied kingfisher X_i according to Equation (10).
    end
    if $X_i \in POP_p(\kappa)$ **then**
      Update the position of pied kingfisher X_i according to Equation (31).
    end
  end
  Compute the current fitness of pied kingfisher fit_{pko}, and $fit_{best} \leftarrow fit_{pko}$.
  Set the optimal position R_{opt} as the position of fit_{best}.
end
return R_{opt}

Table 1
The properties of the 16 datasets.

No.	Datasets	Samples	Attributes	Classes
1	Breast Cancer Coimbra	116	9	2
2	Glass1	214	9	2
3	Plrx	182	12	2
4	Speaker Accent Recognotion	329	12	6
5	Heart	270	13	2
6	Wine	178	13	3
7	Climate Model Simulation Crashes	540	18	2
8	Messidor features	1151	19	2
9	Parkinsons	195	21	2
10	Wpbc	194	21	2
11	Wdbc	569	23	2
12	Audit	776	26	2
13	Ionosphere	351	33	2
14	Hill	606	100	2
15	Colon	62	2000	2
16	Toxicity	171	1203	2

the accuracy of each class independently, thus mitigating the bias introduced by imbalanced datasets.

Five attribute reduction algorithms are selected for comparison with 3PKO-ANRAR. The rationale for including these algorithms stems from their methodological similarities, shared focus on neighborhood rough set models, ability to handle heterogeneous datasets, and applicability to problems involving both attribute reduction and classification accuracy. A brief introduction to the comparison algorithms and the reasons for their selection for evaluation is provided below.

(1) The HARNRS algorithm is included as it is a well-established method for heterogeneous feature subset selection, using the η-neighborhood rough set model to evaluate feature discriminative ability (Hu, Yu et al., 2008). This algorithm is suitable for datasets containing both numerical and categorical attributes, making it a strong benchmark for evaluating the flexibility and adaptability of the proposed algorithm. Comparing against HARNRS demonstrates whether 3PKO-ANRAR can achieve comparable or better performance in handling heterogeneous feature spaces.

(2) The Far-VPKNN algorithm applies k-nearest neighbor relationships to neighborhood rough sets, handling both numerical and categorical attributes through granulation and approximation methods (Hu, Liu et al., 2008). The model is used to assess the importance of mixed features and constructs a greedy attribute reduction algorithm. Its greedy attribute reduction approach is particularly relevant for evaluating the efficiency and effectiveness of the proposed algorithm's search mechanism (i.e., the three-way Pied Kingfisher Optimizer). By comparing with Far-VPKNN, we can assess whether the proposed algorithm's attribute subset search mechanism outperforms traditional greedy methods.

(3) The FSMRI algorithm is a feature selection method based on adaptive neighborhood rough sets that rely on sample neighborhood label distributions (Qu et al., 2023). This algorithm quantifies the correlation between features and decisions using Rough Mutual Information and applies the Maximal Relevance and Minimal Redundancy principle for feature selection. Since 3PKO-ANRAR also uses adaptive neighborhood rough sets and aims to reduce time complexity while improving classification accuracy, comparison with FSMRI can assess the balance between running time and efficiency in the proposed algorithm.

(4) The NNRS algorithm is an attribute reduction method based on the k-nearest neighborhood rough set (k-NNRS) model (Wang et al., 2019). This model combines the advantages of the δ-neighborhood and k-nearest neighbors, effectively addressing non-homogeneous sample distributions, which makes it an excellent benchmark for evaluating how well the adaptive neighborhood radius mechanism in 3PKO-ANRAR handles datasets with non-homogeneous distributions or uneven sample densities. By comparing NNRS, we can assess whether the proposed approach improves upon the robustness and flexibility of traditional neighborhood-based methods in handling diverse distributions.

(5) The HARCD algorithm is a feature selection method based on the neighborhood distinguishability matrix, designed to select effective feature subsets in heterogeneous datasets (Wang et al., 2018). This algorithm considers both sample consistency and the ability to distinguish between different decision outcomes, making it highly relevant for evaluating classification performance. By comparing with HARCD, we can directly assess the efficacy of 3PKO-ANRAR in selecting optimal feature subsets and improving classification results.

4.3. Experimental results and analysis

In this section, we demonstrate the effectiveness of the proposed attribute selection algorithm through comprehensive comparative experiments. The evaluation is based on three key metrics: the running time required to generate the reducts, the size of the reducts, the length of the reducts, and the classification accuracy and balanced accuracy of the reducts on the specific classifier.

The comparison of the running time of different attribute reduction algorithms is shown in Table 2, where our algorithm shows competitive performance, ranking favorably against the other five algorithms across multiple datasets. Notably, the proposed algorithm exhibits a significantly lower average runtime of 755.4 s compared to HARNRS, Far-VPKNN, and HARCD. It also outperforms NNRS on most datasets, though FSMRI achieves the fastest average runtime at 14.5 s. These results indicate that 3PKO-ANRAR performs competitively, achieving

Table 2
The reduction time of 5 comparison algorithms(s).

No.	Datasets	3PKO-ANRAR	HARNRS	Far-VPKNN	FSMRI	NNRS	HARCD
1	bcc	2	7	6	1	5	3
2	glass1	4	52	20	1	9	29
3	plrx	5	27	15	1	18	17
4	sar	26	113	91	1	19	57
5	heart	23	112	68	1	43	45
6	wine	7	35	33	1	16	18
7	cmsc	156	542	591	1	47	366
8	mf	621	9049	1996	2	1312	4981
9	parkinsons	19	101	74	1	48	71
10	wpbc	32	265	176	1	105	151
11	wdbc	36	13	12	1	17	17
12	audit	732	4060	1387	2	1421	2921
13	ionosphere	235	1036	899	1	401	571
14	hill	9152	26 912	13 172	11	5921	18 962
15	tc	635	1358	3543	204	675	3201
16	colon	402	1513	812	48	241	1698
	average	755.4	2824.7	1430.1	17.4	643.6	2069.2

shorter runtimes than methods such as HARNRS, Far-VPKNN, and HARCD, while being slightly slower than FSMRI. This difference is attributed to the iterative nature of PKO and the computational overhead introduced by the adaptive scaling factor, which results in slightly longer runtimes for 3PKO-ANRAR. Although FSMRI achieves the fastest average runtime, it compromises classification accuracy and reduction quality. This makes 3PKO-ANRAR a more advantageous choice for scenarios where computational resources allow for slightly higher runtimes in exchange for better overall performance.

Table 3 presents the reduction lengths across 16 datasets using various comparison algorithms. Our proposed 3PKO-ANRAR algorithm generally achieves shorter reduction lengths, demonstrating its efficiency in attribute reduction. Table 4 details the specific attributes selected by each algorithm for the same datasets, highlighting differences in attribute selection across methods. Our algorithm consistently produces more concise attribute sets while maintaining essential information across datasets. Regarding reduction set size, 3PKO-ANRAR demonstrates significant advantages by generating smaller feature subsets without compromising classification accuracy, reflecting its ability to balance the trade-off between dataset simplification and predictive power. In contrast, algorithms such as NNRS tend to generate larger subsets due to less sophisticated reduction strategies, which may include redundant features. While FSMRI produces small subsets, its reliance on Fisher Score dimensionality reduction sometimes overlooks critical feature interactions, leading to a loss in classification accuracy. The ability of 3PKO-ANRAR to consistently achieve smaller subsets while maintaining or improving classification accuracy underscores its effectiveness in simplifying datasets, reducing computational costs for downstream tasks, and enhancing model interpretability.

Figs. 6 and 7 displays the accuracy and the balanced accuracy of various algorithms on KNN and Decision Tree classifiers. The proposed algorithm consistently outperforms other algorithms, particularly with the Decision Tree, where it achieves the highest accuracy and balanced accuracy across most datasets.

Tables 5 and 6 show the results of different algorithms on the Decision Tree classifier. Our algorithm leads in average accuracy (0.7632) and balanced accuracy (0.7255), indicating strong performance in both overall prediction and handling class imbalances. Tables 7 and 8 detail the results of different algorithms on the KNN classifier. Here, our algorithm once again performs excellently, with an average accuracy of 0.7848 and a balanced accuracy of 0.7215, surpassing other methods in maintaining predictive stability across different classes.

It is interesting that even when the proposed algorithm is not the leader in terms of raw accuracy, it often ranks highest in terms of balanced accuracy, indicating its robustness in managing different class distributions. This highlights the algorithm's adaptive capabilities, making it effective for complex datasets with uneven class distributions. In terms of classification accuracy, 3PKO-ANRAR consistently outperforms the comparison methods across a wide range of datasets. This robust performance can be attributed to the adaptive neighborhood rough set model, which effectively captures local data distributions, and the well-designed fitness function that emphasizes both attribute dependency and minimal redundancy. Unlike traditional methods such as Far-VPKNN or NNRS, which rely on fixed-radius or greedy strategies, 3PKO-ANRAR dynamically adapts to the characteristics of each dataset, avoiding premature convergence through its three-way decision mechanism. Although FSMRI and HARCD are more efficient in certain respects, their predictive accuracy often lags behind due to their reliance on simpler or less flexible optimization processes.

The Wilcoxon test is a non-parametric statistical method used to compare two paired groups. When the data cannot be assumed to be normally distributed, it can serve as an alternative to the paired t-test. It evaluates whether the median differences between pairs of observations are zero, making it particularly useful for assessing performance differences in computational experiments.

In Table 9, the Wilcoxon test results compare the 3PKO-ANRAR algorithm with other algorithms across two classifiers: K-Nearest Neighbors (KNN) and Decision Tree (DT). The threshold was set to 0.05 in the experiment. This table evaluates two metrics: accuracy and balanced accuracy, offering insights into the comparative effectiveness of the algorithms. For the KNN classifier, the P-values remain below 0.05, including FSMRI, NNRS, and RAW, in terms of accuracy, and the same pattern is observed in terms of balanced accuracy. Thus, the results demonstrate that both the accuracy and balanced accuracy of 3PKO-ANRAR are significant distinct from those of the three algorithms. Regarding the DT classifier, most P-values are below 0.05, with the exception of HARNRS and HARCD in terms of accuracy. Similarly, in terms of balanced accuracy, the P-values of HARNRS, Far-VPKNN, FSMRI and HARCD are all below 0.05. This indicates that 3PKO-ANRAR exhibits significant differences in accuracy compared to approaches other than HARNRS and HARCD, and exhibits significant differences in balanced accuracy compared to approaches other than NNRS, HARCD, and RAW. These statistical findings underscore the robust classification performance of 3PKO-ANRAR.

Fig. 8 provides a visual representation of 9, clearly depicting the performance comparison of existing algorithms with the proposed algorithm across two classifiers. The chart is presented as a bar graph, where each bar is divided into segments representing the p-values of different algorithms. The height of each segment indicates the percentage of that algorithm's p-value relative to the total p-value of all algorithms. A smaller segment proportion suggests a more significant performance difference between the algorithms.

5. Conclusions and future work

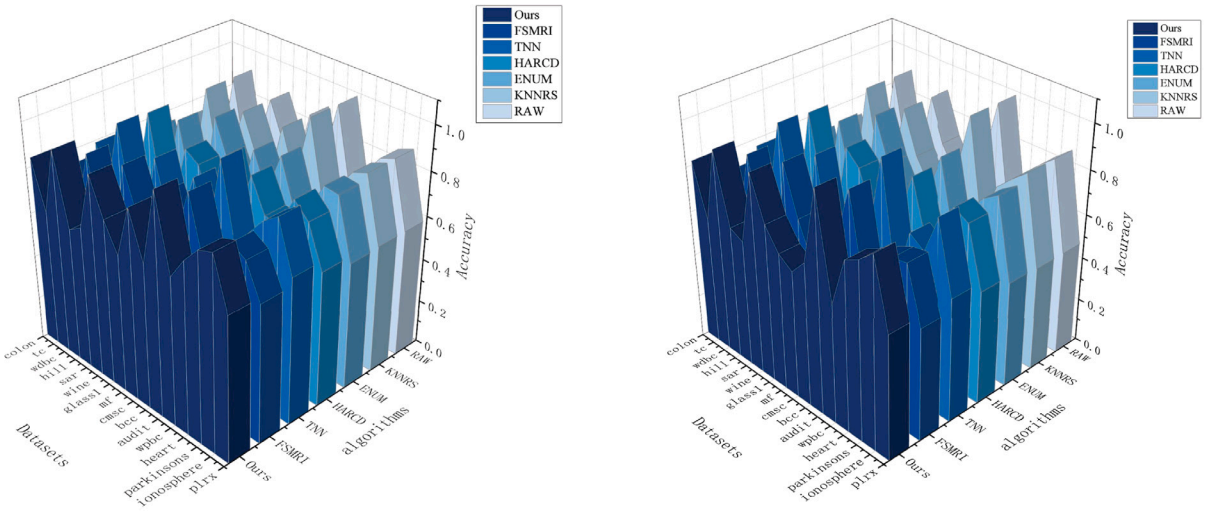
This paper proposed a method for attribute reduction based on adaptive neighborhood rough sets and an improved PKO algorithm. This approach addresses the challenge of setting neighborhood radii in traditional rough sets and enhances the performance of attribute reduction. By incorporating sample distribution information to construct adaptive neighborhood relations and setting a scaling factor, the approach achieves automatic adjustment of neighborhood radii, thereby improving the adaptability to various data distributions. Furthermore, leveraging the robust search capabilities of the PKO algorithm and the dependency of adaptive neighborhood rough sets, a fitness function is designed to evaluate the quality of reducts effectively and identify the optimal reduct. To avoid local optima, a three-way population division mechanism and local perturbation strategy are introduced, significantly enhancing the robustness of our approach. Despite these achievements, the proposed approach has some limitations. Firstly, the adaptive radius of the neighborhood depends on the sample distribution, which may

Table 3
The length of the reduction results of 5 comparison algorithms.

No.	Datasets	3PKO-ANRAR	HARNRS	Far-VPKNN	FSMRI	NNRS	HARCD	RAW
1	bcc	3	2	3	2	9	4	9
2	glass1	2	4	1	2	9	4	9
3	plrx	1	3	1	1	12	2	12
4	sar	7	3	3	1	12	4	12
5	heart	8	5	3	2	13	3	13
6	wine	6	3	2	2	13	4	13
7	cmsc	5	2	2	1	18	3	18
8	mf	4	7	3	2	19	4	19
9	parkinsons	5	3	1	2	22	5	22
10	wpbc	4	2	1	2	33	2	33
11	wdbc	5	3	3	1	23	6	23
12	audit	11	3	1	3	22	3	22
13	ionosphere	5	3	3	1	33	2	33
14	hill	10	3	2	7	100	7	100
15	tc	4	2	2	6	1206	2	1206
16	colon	3	6	4	1	2000	2	2000

Table 4
The length of the reduction results of 5 comparison algorithms.

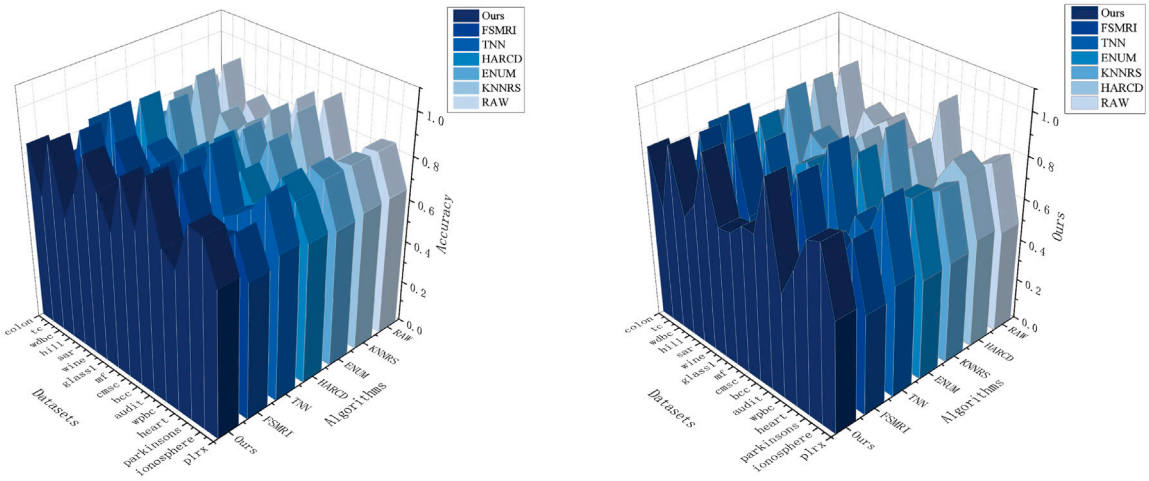
No.	Datasets	3PKO-ANRAR	HARNRS	FarVPKNN	FSMRI	NNRS	HARCD	RAW
1	bcc	1,3,8	1,3	1,4,8	1,3	all	1,2,3,7	9
2	glass1	1,4	1,3,4,9	8	5,6	all	1,3,4,6	9
3	plrx	12	3,6,12	12	1	all	1,9	12
4	sar	1,3,4,5,6,10,11	1,3,10	5,6,12	3	all	1,2,8,10	12
5	heart	1,2,3,5,9,11,12,13	3,4,9,10,11	3,10,13	8,13	all	7,10,12	13
6	wine	1,4,5,7,10,13	7,10,13	7,11	7,10	all	1,6,8,13	13
7	cmsc	1,2,14,16,18	2,14	1,3	3	all	1,2,6	18
8	mf	3,7,10,11	3,6,9,10,11,15,18	3,7,16	11,18	all	1,7,8,16	19
9	parkinsons	5,18,19,20,21	1,19,20	5	5,6	all	1,7,16,19,20	22
10	wpbc	1,5,16,31	1,15	23	11,28	all	1,12	33
11	wdbc	22,23,24,25,28	22,23,30	23,26,29	8	all	3,4,21,22,23,24	23
12	audit	11,12,13,14,15,17,18,19, 21,22,23,24,25,26	14,19,22	26	4,14,18	Exceptfor 2,4,13,25	2,4,14	26
13	ionosphere	3,9,26,36	2,31,32	4,5,17	33	all	1,7	33
14	hill	1,2,5,16,18,44,52,72,89,98	2,14,15	25,73	32,42,55,64,65,67,71	all	22,43,8,72,74,83,89	100
15	tc	204,430,726,1096	517 571	430 442	67,204,264,343,798,1000	all	41,464	1206
16	colon	95,258,1411	1293,1673	95,258,1047,1411	1909	all	72,781,1187,1231, 1241,1539	2000



(a) Accuracy on KNN (b) Accuracy on Decision Tree

Fig. 6. The accuracy of different algorithms on two classifiers.

restrict the algorithm's performance when dealing with datasets featuring highly uneven distributions. Secondly, although the improved PKO algorithm increases the search efficiency, its computational complexity and memory consumption could become requirements when handling large-scale datasets. Future work could focus on optimizing the algorithm to accommodate a wider range of data distributions and exploring more efficient computational strategies for large-scale datasets. Additionally, future research could consider parallel and distributed



(a) Balanced accuracy on KNN (b) Balanced accuracy on Decision Tree

Fig. 7. The balanced accuracy of different algorithms on two classifiers.

Table 5
Comparison results of accuracy for different algorithms on the DT classifier.

No.	Datasets	3PKO-ANRAR	HARNRS	Far-VPKNN	FSMRI	NNRS	HARCD	RAW
1	bcc	0.6719	**0.7181**	0.6236	0.7104	0.6946	0.6552	0.6977
2	glass1	0.7488	0.7543	0.6254	0.7009	0.7389	**0.8061**	0.7359
3	plrx	0.6646	0.5901	**0.6716**	0.6275	0.5911	0.6249	0.5816
4	sar	0.6442	0.5145	0.5115	0.3848	0.6634	0.5837	**0.6635**
5	heart	**0.7858**	0.7213	0.7607	0.7137	0.7622	0.7182	0.7459
6	wine	0.9127	**0.9343**	0.8543	0.8915	0.8944	0.8461	0.8865
7	cmsc	0.8694	0.8551	0.8557	0.8542	**0.9109**	0.8959	**0.9109**
8	mf	0.6341	0.6237	**0.6478**	0.5312	0.6137	0.5922	0.6166
9	parkinsons	0.8545	**0.8611**	0.8351	0.8099	0.8435	0.8536	0.8476
10	wpbc	0.6836	0.6854	0.6733	0.6654	0.6767	**0.7282**	0.6813
11	wdbc	**0.9324**	0.8324	0.9246	0.8628	0.9165	0.9117	0.9153
12	audit	0.9981	0.8937	0.9945	0.9364	**0.9999**	0.8693	0.9919
13	ionosphere	0.8845	0.8457	0.8819	0.7861	0.8753	0.7988	**0.8861**
14	hill	**0.6081**	0.5542	0.5492	0.5208	0.5902	0.5629	0.5809
15	tc	0.6209	0.5335	**0.6433**	0.5649	0.5751	0.5928	0.5715
16	colon	**0.8333**	0.7466	0.7883	0.7327	0.7428	0.7145	0.7388
	average	**0.7717**	0.7289	0.7401	0.7059	0.7556	0.7346	0.7533

Table 6
Comparison results of balance accuracy for different algorithms on the DT classifier.

No.	Datasets	3PKO-ANRAR	HARNRS	Far-VPKNN	FSMRI	NNRS	HARCD	RAW
1	bcc	0.6562	**0.7067**	0.6237	0.6956	0.7011	0.6464	0.6942
2	glass1	0.7201	**0.7331**	0.4843	0.682	0.7102	0.7745	0.7209
3	plrx	**0.5782**	0.4827	0.5718	0.5063	0.4869	0.5234	0.4864
4	sar	**0.5872**	0.3937	0.3795	0.216	0.5725	0.47	0.5697
5	heart	**0.7672**	0.7079	0.7454	0.7147	0.7535	0.7174	0.7536
6	wine	0.9161	**0.9183**	0.8592	0.8971	0.9022	0.8377	0.8973
7	cmsc	0.5911	0.5439	0.6023	0.501	0.6681	0.648	**0.6777**
8	mf	0.6344	0.6126	**0.6538**	0.5148	0.6058	0.5883	0.6089
9	parkinsons	0.8086	0.8002	0.7014	0.7545	0.8157	0.8209	**0.8218**
10	wpbc	0.5473	0.5603	0.5076	0.5326	0.5579	**0.6189**	0.5614
11	wdbc	0.9181	0.8282	**0.9273**	0.8492	0.9214	0.9115	0.9087
12	audit	**0.9997**	0.8902	0.9981	0.9299	0.9989	0.8503	0.9996
13	ionosphere	0.8728	0.8425	0.8729	0.7497	0.8631	0.7842	**0.8735**
14	hill	**0.6109**	0.5477	0.5384	0.5111	0.5906	0.5637	0.5812
15	tc	0.5863	0.4769	**0.5912**	0.5118	0.5038	0.5226	0.5133
16	colon	**0.8217**	0.7118	0.7683	0.7128	0.7181	0.665	0.7173
	average	**0.7263**	0.6728	0.6759	0.6424	0.7106	0.6841	0.7116

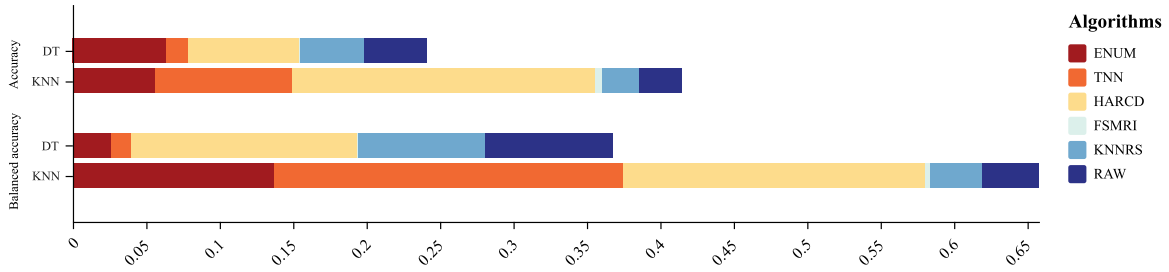


Fig. 8. The visualization of the wilcoxon test results.

Table 7
Comparison results of average accuracy for different comparative algorithms on the KNN classifier.

No.	Datasets	3PKO-ANRAR	HARNRS	Far-VPKNN	FSMRI	NNRS	HARCD	RAW
1	bcc	0.7295	0.7072	0.7117	0.7093	0.5291	**0.7531**	0.5291
2	glass1	0.6773	0.7725	0.6348	0.7892	0.6728	**0.8021**	**0.8021**
3	plrx	**0.6884**	0.6457	0.6855	0.6407	0.6612	0.6622	0.6612
4	sar	0.7306	0.5731	0.5329	0.3953	**0.8013**	0.6197	**0.8013**
5	heart	0.6654	0.6987	**0.7623**	0.7162	0.6481	0.7408	0.6481
6	wine	**0.9312**	0.7132	0.8518	0.9204	0.7216	0.6903	0.7216
7	cmsc	**0.9291**	0.8924	0.8971	0.8954	0.9179	0.9076	0.9179
8	mf	**0.6548**	0.6515	0.6514	0.5171	0.6489	0.5657	0.6489
9	parkinsons	**0.8865**	0.8722	0.8166	0.7383	0.8339	0.8856	0.8339
10	wpbc	0.7198	**0.7223**	0.7192	0.7196	0.7129	0.7103	0.7129
11	wdbc	0.9017	0.8612	**0.9337**	0.8941	0.9274	0.9256	0.9274
12	audit	**0.9897**	0.8929	0.9983	0.9314	0.9632	0.8231	0.9632
13	ionosphere	0.8734	0.8433	**0.9092**	0.8153	0.8436	0.8136	0.8436
14	hill	**0.5886**	0.5556	0.5559	0.5386	0.5542	0.5563	0.5545
15	colon	**0.8372**	0.7273	0.8337	0.7223	0.7982	0.6653	0.7982
16	tc	0.6292	0.5987	0.6283	0.6378	0.5456	**0.6329**	0.5456
	average	**0.7848**	0.7348	0.7681	0.7246	0.7443	0.7346	0.7443

Table 8
Comparison results of average balance accuracy for different algorithms on the KNN classifier.

No.	Datasets	3PKO-ANRAR	HARNRS	Far-VPKNN	FSMRI	NNRS	HARCD	RAW
1	bcc	0.7126	0.7231	0.7054	0.7076	0.5121	**0.7496**	0.5121
2	glass1	0.6387	0.7449	0.4929	0.6181	0.7675	0.7587	**0.7675**
3	plrx	**0.5373**	0.4847	0.5357	0.4794	0.4907	0.5259	0.4907
4	sar	0.6775	0.4289	0.3852	0.1953	**0.7652**	0.4934	0.765
5	heart	0.6525	0.6841	**0.7492**	0.7054	0.6329	0.7263	0.6329
6	wine	**0.9439**	0.7014	0.8725	0.9306	0.7178	0.6768	0.7178
7	cmsc	**0.7107**	0.5223	0.5194	0.4992	0.5983	0.6207	0.5983
8	mf	0.6475	**0.6587**	0.6573	0.5176	0.6526	0.5674	0.6526
9	parkinsons	0.8213	0.8113	0.6974	0.6368	0.7544	**0.8404**	0.7544
10	wpbc	0.5089	**0.5689**	0.5257	0.5075	0.5313	0.5464	0.5316
11	wdbc	0.8889	0.8425	**0.9266**	0.8865	0.9182	0.9139	0.9182
12	audit	**0.9889**	0.9028	0.9978	0.9339	0.9543	0.8105	0.9544
13	ionosphere	0.8452	0.8235	**0.8921**	0.7852	0.7921	0.7903	0.7921
14	hill	**0.5871**	0.5539	0.5556	0.5368	0.5558	0.5538	0.5558
15	colon	**0.8253**	0.6824	0.8232	0.6712	0.7404	0.5749	0.7404
16	tc	**0.5609**	0.5034	0.5583	0.5564	0.4581	0.5098	0.4581
	average	**0.7215**	0.6633	0.6809	0.6355	0.6775	0.6662	0.6775

Table 9
The wilcoxon test results of 3PKO-ANRAR and other five comparative algorithms.

Classifier	Metric	HARNRS	Far-VPKNN	FSMRI	NNRS	HARCD	RAW
KNN	Acc	0.0561	0.0931	0.0042	0.0257	0.2067	0.0286
	BA	0.1371	0.2375	0.0034	0.0356	0.2058	0.0386
DT	Acc	0.0634	0.0151	0.0007	0.0429	0.0765	0.0426
	BA	0.0263	0.0135	0.0006	0.0867	0.1539	0.0867

implementations to further enhance processing capabilities.

CRediT authorship contribution statement

Wenjing Qiu: Conceptualization, Methodology, Writing – original draft, Software, Validation. **Caihui Liu:** Conceptualization, Methodology, Writing – review & editing, Validation, Supervision . **Bowen Lin:** Data curation, Software. **Xiying Chen:** Data curation, Software. **Duoqian Miao:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The research is supported by the National Natural Science Foundation of China under Grant No. 62166001, Jiangxi Province Natural Science Foundation of China under Grant No. 20232ACB202013.

Data availability

Data will be made available on request.

References

Bouaouda, A., Hashim, F. A., Sayouti, Y., & Hussien, A. G. (2024). Pied kingfisher optimizer: a new bio-inspired algorithm for solving numerical optimization and industrial engineering problems. *Neural Computing and Applications*, 1–59.

Chen, H., Li, T., Fan, X., & Luo, C. (2019). Feature selection for imbalanced data based on neighborhood rough sets. *Information Sciences*, *483*, 1–20.

Chen, X., Liu, C., Lin, B., Lai, J., & Miao, D. (2024). AHA-3WKM: The optimization of K-means with three-way clustering and artificial hummingbird algorithm. *Information Sciences*, *672*, Article 120661.

Chen, Y., Zhu, Q., & Xu, H. (2015). Finding rough set reducts with fish swarm algorithm. *Knowledge-Based Systems*, *81*, 22–29.

Guo, D., Xu, W., Ding, W., Yao, Y., Wang, X., Pedrycz, W., & Qian, Y. (2024). Concept-cognitive learning survey: Mining and fusing knowledge from data. *Information Fusion*, *109*, Article 102426.

Gupta, S., & Gupta, S. (2024). Fitness and historical success information-assisted binary particle swarm optimization for feature selection. *Knowledge-Based Systems*, Article 112699.

Hu, Q., Liu, J., & Yu, D. (2008). Mixed feature selection based on granulation and approximation. *Knowledge-Based Systems*, *21*(4), 294–304.

Hu, Q., Yu, D., Liu, J., & Wu, C. (2008). Neighborhood rough set based heterogeneous feature subset selection. *Information Sciences*, *178*(18), 3577–3594.

Ibrahim, R. A., Abd Elaziz, M., Oliva, D., & Lu, S. (2020). An improved runner-root algorithm for solving feature selection problems based on rough sets and neighborhood rough sets. *Applied Soft Computing*, *97*, Article 105517.

Liu, Q., Cai, M., & Li, Q. (2024). Supervised spectral feature selection with neighborhood rough set. *Applied Soft Computing*, *165*, Article 112111.

Liu, K., Fu, Y., Wu, L., Li, X., Aggarwal, C., & Xiong, H. (2021). Automated feature selection: A reinforcement learning perspective. *IEEE Transactions on Knowledge and Data Engineering*, *35*(3), 2272–2284.

Liu, C., Lin, B., & Miao, D. (2024). A novel adaptive neighborhood rough sets based on sparrow search algorithm and feature selection. *Information Sciences*, *679*, Article 121099.

Qu, K., Xu, J., Han, Z., & Xu, S. (2023). Maximum relevance minimum redundancy-based feature selection using rough mutual information in adaptive neighborhood rough sets. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(14), 17727–17746.

Sewwandi, M. A. N. D., Li, Y., & Zhang, J. (2023). A class-specific feature selection and classification approach using neighborhood rough set and K-nearest neighbor theories. *Applied Soft Computing*, *143*, Article 110366.

Shu, T., Lin, Y., & Guo, L. (2024). Online hierarchical streaming feature selection based on adaptive neighborhood rough set. *Applied Soft Computing*, *152*, Article 111276.

Sun, L., Chen, Y., Ding, W., Xu, J., & Ma, Y. (2023). AMFSA: Adaptive fuzzy neighborhood-based multilabel feature selection with ant colony optimization. *Applied Soft Computing*, *138*, Article 110211.

Tawhid, M. A., & Ibrahim, A. M. (2020). Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. *International Journal of Machine Learning and Cybernetics*, *11*(3), 573–602.

Tsai, C.-F., Chen, K.-C., & Lin, W.-C. (2024). Feature selection and its combination with data over-sampling for multi-class imbalanced datasets. *Applied Soft Computing*, *153*, Article 111267.

Wan, J., Chen, H., Yuan, Z., Li, T., Yang, X., & Sang, B. (2021). A novel hybrid feature selection method considering feature interaction in neighborhood rough set. *Knowledge-Based Systems*, *227*, Article 107167.

Wang, C., He, Q., Shao, M., & Hu, Q. (2018). Feature selection based on maximal neighborhood discernibility. *International Journal of Machine Learning and Cybernetics*, *9*, 1929–1940.

Wang, F., Liang, J., & Qian, Y. (2013). Attribute reduction: a dimension incremental strategy. *Knowledge-Based Systems*, *39*, 95–108.

Wang, C., Shi, Y., Fan, X., & Shao, M. (2019). Attribute reduction based on k-nearest neighborhood rough sets. *International Journal of Approximate Reasoning*, *106*, 18–31.

Wang, C., Wang, C., Qian, Y., & Leng, Q. (2024). Feature selection based on weighted fuzzy rough sets. *IEEE Transactions on Fuzzy Systems*.

Wang, N., & Zhao, E. (2024). A new method for feature selection based on weighted k-nearest neighborhood rough set. *Expert Systems with Applications*, *238*, Article 122324.

Xia, D., Wang, G., Zhang, Q., Yang, J., & Xia, S. (2024). Three-way approximations fusion with granular-ball computing to guide multi-granularity fuzzy entropy for feature selection. *IEEE Transactions on Fuzzy Systems*.

Xu, W., & Bu, Q. (2024). Feature selection using generalized multi-granulation dominance neighborhood rough set based on weight partition. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Xu, W., Guo, D., Mi, J., Qian, Y., Zheng, K., & Ding, W. (2023). Two-way concept-cognitive learning via concept movement viewpoint. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(10), 6798–6812.

Xu, W., Yuan, Z., & Liu, Z. (2023). Feature selection for unbalanced distribution hybrid data based on k-nearest neighborhood rough set. *IEEE Transactions on Artificial Intelligence*, *5*(1), 229–243.

Yang, J., Liu, Z., Xia, S., Wang, G., Zhang, Q., Li, S., & Xu, T. (2024). 3WC-GBNRS++: A novel three-way classifier with granular-ball neighborhood rough sets based on uncertainty. *IEEE Transactions on Fuzzy Systems*.

Yao, Y. (2011). The superiority of three-way decisions in probabilistic rough set models. *Information Sciences*, *181*(6), 1080–1096.

Yin, T., Chen, H., Wan, J., Zhang, P., Horng, S.-J., & Li, T. (2024). Exploiting feature multi-correlations for multilabel feature selection in robust multi-neighborhood fuzzy β covering space. *Information Fusion*, *104*, Article 102150.

Yuan, K., Miao, D., Pedrycz, W., Ding, W., & Zhang, H. (2024). Ze-HFS: Zentropy-based uncertainty measure for heterogeneous feature selection and knowledge discovery. *IEEE Transactions on Knowledge and Data Engineering*.

Yuan, K., Miao, D., Yao, Y., Zhang, H., & Zhao, X. (2023). Feature selection using Zentropy-based uncertainty measure. *IEEE Transactions on Fuzzy Systems*.

Yuan, K., Xu, W., & Miao, D. (2024). A local rough set method for feature selection by variable precision composite measure. *Applied Soft Computing*, *155*, Article 111450.

Zhou, P., Hu, X., Li, P., & Wu, X. (2019). Online streaming feature selection using adapted neighborhood rough set. *Information Sciences*, *481*, 258–279.