Computers in Biology and Medicine 166 (2023) 107538



Contents lists available at ScienceDirect

Computers in Biology and Medicine

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





A new population initialization of metaheuristic algorithms based on hybrid fuzzy rough set for high-dimensional gene data feature selection

Xuanming Guo [a], Jiao Hu [a], Helong Yu [b], Mingjing Wang [c,*], Bo Yang [a,**]

[a] College of Computer Science and Technology, Jilin University, Changchun, 130012, China
[b] College of Information Technology, Jilin Agricultural University, Changchun, 130118, China
[c] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou, 325000, China

ARTICLE INFO

Keywords:
Population initialization
Hybrid fuzzy rough set
Whale optimization algorithm
Gene data feature selection
Multiclass classification

ABSTRACT

In the realm of modern medicine and biology, vast amounts of genetic data with high complexity are available. However, dealing with such high-dimensional data poses challenges due to increased processing complexity and size. Identifying critical genes to reduce data dimensionality is essential. The filter-wrapper hybrid method is a commonly used approach in feature selection. Most of these methods employ filters such as MRMR and ReliefF, but the performance of these simple filters is limited. Rough set methods, on the other hand, are a type of filter method that outperforms traditional filters. Simultaneously, many studies have pointed out the crucial importance of good initialization strategies for the performance of the metaheuristic algorithm (a type of wrapper-based method). Combining these two points, this paper proposes a novel filter-wrapper hybrid method for high-dimensional feature selection. To be specific, we utilize the variant of bWOA (binary Whale Optimization Algorithm) based on Hybrid Fuzzy Rough Set to perform attribute reduction, and the reduced attributes are used as prior knowledge to initialize the population. We then employ metaheuristics for further feature selection based on this initialized population. We conducted experiments using five different algorithms on 14 UCI datasets. The experiment results show that after applying the initialization method proposed in this article, the performance of five enhanced algorithms, has shown significant improvement. Particularly, the improved bMFO using our initialization method: fuzzy_bMFO outperformed six currently advanced algorithms, indicating that our initialization method for metaheuristic algorithms is suitable for high-dimensional feature selection tasks.

1. Introduction

DNA microarray datasets provide information about gene expression in cells and tissues. This kind of dataset plays a crucial role in current research in the fields of bioinformatics and machine learning. By utilizing these datasets, researchers can make diagnoses for diseases and distinguish different types of tumors [1].

However, these raw data often have a high number of dimensions, while the corresponding sample number is low. Using this kind of data for feature selection and disease prediction can lead to a high false positive rate. At the same time, not every dimension in DNA microarray datasets contributes positively to disease classification; many features are redundant [2]. Using the raw data for disease classification can increase computational complexity and reduce classification accuracy [3, 4]. To avoid "curse of dimensionality" [5], researchers resort to feature selection methods to study DNA microarrays. Feature selection refers to the process of eliminating variables that are irrelevant to downstream tasks (such as classification) [6], thereby improving model performance and reducing computation time.

There are various methods for feature selection. Based on the role of the classifier in the feature selection task, we can roughly categorize these methods into three different types. The first type of method is the filter approach. In this approach, the feature extraction process does not involve the use of a classifier; instead, it solely relies on the characteristics of the data itself. As a result, this kind of methods consume less time, thus suitable for high-dimensional data. Some advanced filter algorithms are applied to feature selection in high-dimensional data, such as: MRMR [7], ReliefF [8], CFS [9], and FCBF [10].

* Corresponding author.
** Corresponding author.
E-mail addresses: guoxm21@mails.jlu.edu.cn (X. Guo), hujiao22@mails.jlu.edu.cn (J. Hu), yuhelong@jlau.edu.cn (H. Yu), wangmingjing.style@gmail.com (M. Wang), ybo@jlu.edu.cn (B. Yang).

https://doi.org/10.1016/j.compbiomed.2023.107538
Received 7 August 2023; Received in revised form 6 September 2023; Accepted 28 September 2023
Available online 4 October 2023
0010-4825/© 2023 Elsevier Ltd. All rights reserved.

Another type of method in feature selection is the wrapper approach. Unlike the filter method, the wrapper approach requires the use of a classifier to evaluate the quality of selected features during the feature selection task. The choice of classifier can be diverse, for example, one can use SVM [11,12], employ KNN [13,14], or utilize artificial neural networks (ANN) [15,16]. While wrapper methods tend to result in higher-quality selected features, they also come with increased computational complexity. It's not feasible to analyze the classification performance for every feature subset using above classifiers. Therefore, heuristic algorithms are often used instead of exhaustive methods and random search algorithms [17]. Metaheuristics are a class of the heuristic algorithms which are designed to tackle a wide range of problems [18]. Over the past 20 years, metaheuristics have demonstrated their efficacy and success in addressing complex, extensive issues across various domains such as multilevel threshold image segmentation [105,106], feature selection [93,107], global optimization [108], paraquat poisoning patient analysis [109], pulmonary hypertension diagnoses [110], medical diagnosis [111,112], and engineering design [113].

Metaheuristic algorithms usually draw inspiration from physics [19] and nature [20]. They have been tested and proven capable of finding high-quality solutions within a limited time frame. For example, inspired by the echolocation behavior of microbats, Yang et al. introduced the Bat Algorithm [21]. Building upon the concept of swarm intelligence, Rao et al. proposed JAYA algorithm [22]. Rashedi et al. proposed the GSA [23], leveraging mathematical modeling based on Newton's law of gravity and motion. Mirjalili et al. introduced the Ant Lion Optimizer (ALO) [24], drawing inspiration from the hunting mechanism of antlions. Additionally, Mirjalili et al. proposed the Moth-Flame Optimization (MFO) algorithm [25]. This algorithm takes cues from the navigation mechanism of moths during night time. Li et al. drew inspiration from the propagation wave of slime and introduced the Slime Mould Algorithm (SMA) [26,114]. Yang et al. devised the Hunger Games Search (HGS), drawing parallels from animals' hunger-driven activities [27]. Ahmadianfar et al. formulated an efficient optimization algorithm based on RUN [28]. Taking cues from the corporate predation behaviors of animals. Tu et al. introduced the Corporate Predation Algorithm (CPA) [29] as a novel metaheuristic approach. Ahmadianfar et al. introduced INFO [30], an optimizer based on the mean of vectors. Hedari et al. proposed the Harris Hawks Optimization (HHO) [31], drawing inspiration from the hunting behaviors of Harris' hawks. Su et al. presented the RIME algorithm [32], which simulates the growth process of rime ice on surfaces.

The third category of feature selection methods is called hybrid approaches. These approaches combine filter and wrapper together. By doing so, a balance is achieved between the benefits of these two methods: the rapid processing of filters and the superior feature quality of wrappers. Filter-wrapper hybrid methods can be generally categorized into two types. The first type involves using a filter to select a subset of features and then applying a wrapper to further refine the feature selection based on this subset. The other type utilizes filter methods to estimate the features. The evaluation results are then used to guide the population initialization of subsequent metaheuristic wrapper methods.

There are hybrid methods that do not fall in these two categories. However, the core of these methods is still to leverage the advantages of both filter and wrapper techniques to improve feature selection performance [33–35]. Meanwhile, there are other so-called hybrid methods that only combine filter-based methods together or only combine wrapper-based methods together. For example, Angulo et al. combine MRMR and CFS to select feature subset [36], where both MRMR and CFS are typical filter-based methods. Thawkar et al. combined the Butterfly Optimization and Ant Lion Optimizer for breast cancer prediction [37], which can be viewed as wrapper-based methods. These methods can also be called as "hybrid methods" since they indeed combine different algorithms together. However, they failed to integrate the advantages of filter and wrapper methods, so we will not discuss them further here.

In the first hybrid method, the filter method is firstly used to roughly screen the features to obtain a subset of features. Based on this subset, the wrapper method conducts further feature selection. For example: Akadi et al. [38] combine MRMR and GA methodologies. They initially utilize MRMR to filter out significant genes, followed by employing GA to further select the optimal subset. This approach effectively reduces the computational burden of subsequent wrapper methods. Bansal et al. proposed MRMR-PSO [39], which first remove some features of original data using MRMR, then the reduced features are given to PSO for further process. Shreem et al. proposed R-m-GA [40]. This work first uses ReliefF to construct a candidate gene subset, then employs the MRMR method to minimize redundancy, and finally utilizes the GA algorithm to select the genes that have the most significant impact on classification. Salem et al. [41] introduced a new algorithm for cancer disease classification, which first used gain of information for feature screening, then utilized genetic algorithm for feature reduction, and finally utilized genetic planning for classification. Baliarsingh et al. [42] proposed CEPO using Fisher score and ReliefF as filters to screen relevant genes, respectively. Vipsita et al. [43] proposed Jaya-based forest optimization algorithm. They first use the filter ANOVA method to select features, and then they employ the wrapper-based approach named Enhanced Jaya and forest optimization algorithm to find the optimal feature subset. Shukla et al. [44] introduced a new method for gene selection combining MRMR and a teaching-learning based optimization approach for cancer prediction.

In the second method, filters do not filter out unimportant features; instead, they evaluate these features, and the evaluation results are used to guide the population initialization of subsequent metaheuristic wrapper methods. Many studies show that the initial population is critical for metaheuristic algorithms [45–47]. Lots of population initialization strategies for metaheuristics have been proposed over the past few decades [48–50]. The filter-based method is one of them. These methods are useful because they can provide helpful knowledge for metaheuristics [51], thereby enhancing their convergence speed and leading to improved ultimate solutions. For example, Xue et al. [51] proposed that using the ReliefF algorithm for population initialization can improve the performance of PSO. They also combine filter methods such as ReliefF, Information gain and Fisher score together [52] to initialize the population of the metaheuristic algorithm. Similarly, Zhang et al. [53] employ the ReliefF algorithm to effectively rank the initial set of features. Subsequently, they derive feature selection probabilities based on this ranking. Within the Genetic Algorithm (GA) framework, initial individuals are chosen in accordance with these probabilities.

Rough set method [54] is also a filter-based approach [55] for feature selection tasks. Many studies have demonstrated that feature selection methods using rough sets exhibit superior performance compared to other filter approaches [56–59]. Equivalence relations are commonly employed in the field of Classical Rough Set, primarily suited for categorical (nominal) attributes. However, real-life databases often consist of a combination of categorical and numerical attributes. To address this issue, Dubois et al. proposed the Fuzzy Rough Set (FRS) model [60,61], offering a viable solution to overcome the limitation of discretizing continuous data. Furthermore, Hu et al. emphasized that it is inappropriate to use distance to measure the similarity of categorical attributes. Therefore, they proposed a hybrid attribute reduction [62] method based on fuzzy rough sets and information granulation. This method can unify the similarity measurement of categorical attributes and numerical attributes using the same calculation formula.

Although algorithms based on rough set to select optimal subset is advanced, the time consumption to evaluate all subsets is exponential [63]. Therefore, heuristic algorithms must be used. Generally speaking, there are two types of heuristic feature selection methods based on rough sets: one is the greedy approach, and the other is the stochastic approach [64]. Greedy methods start with an empty set and gradually add the currently most significant feature to the candidate feature set.

Alternatively, they can start with a full set and step by step eliminate the features with the least impact on dependency. Region-based attribute significance [65,66] and conditional entropy-based [67,68] attribute significance are often used as guiding heuristics in RS feature selection.

Another heuristic approach for RS feature selection is to use stochastic methods, which means applying metaheuristic algorithms to find minimum reducts. Compared to greedy methods, stochastic methods can provide more robust solutions [69] and are therefore widely used in RS feature selection. For example, Bjorvand applies genetic algorithms (GA) on RS [70]. Emrah Hancer uses differential evolution for attribute reduction on FRS [71]. Wang [72] et al. proposed PSO, and Jensen et al. proposed ACO on RS [73]. Tawhid et al. developed their own binary Whale Optimization (bWOA) algorithm [74] for feature selection and evaluated its performance on rough set theory.

This work presents a novel filter-wrapper hybrid approach for the gene selection task. To be specific, we propose a new population initialization of metaheuristic algorithms based on hybrid fuzzy rough set for high-dimensional gene feature selection. We name our initialization method as "HFRS-initialization".

Our method can be divided into two parts. The first part is population initialization. In this work, we employ a variant of the bWOA to perform attribute reduction based on hybrid fuzzy rough set. This particular variant of WOA exhibits stronger search capability and consumes less time. By taking consideration of both the dependencies and the number of selected features as fitness, we can identify appropriate feature subsets to serve as prior knowledge for the initialization of subsequent Metaheuristic Algorithms (MA) populations.

The second part involves utilizing metaheuristic algorithms with k-nearest neighbors (KNN) as a wrapper for feature selection. We selected five algorithms, namely, bMFO, bBA, bJAYA, bGSA, and bALO, for validation. Experimental results indicate that, after applying our population initialization method, these five algorithms exhibit significant improvements in performance on UCI datasets compared with their original version. Notably, the performance of the bMFO initialized with our method (referred to as fuzzy_ bMFO in the following test) surpasses that of six currently advanced algorithms.

Our work differs from existing research in several key aspects. Firstly, we have introduced a novel filter-wrapper hybrid method. Specifically, we employ the second method mentioned earlier, which involves utilizing a filter for population initialization and subsequently applying a wrapper for feature selection. In contrast to existing approaches that use MRMR or ReliefF as filters, we utilize a hybrid fuzzy rough set as a filter. Some prior research has demonstrated that rough set methods outperform other filter methods in certain tasks [56–59], and we will further validate this in our experiments. Secondly, we apply a variant of the Whale Optimization Algorithm (WOA) to perform attribute reduction in hybrid fuzzy rough set. This method exhibits stronger search capabilities and consumes less time. We also compare the results of attribute reduction obtained using this method with those obtained using other metaheuristic algorithms.

In summary, this work has the following main contributions.

- The variant of WOA is utilized for hybrid fuzzy rough set attribute reduction, enhancing the search capability.
- The reduced attributes are used as the knowledge to initialize population for metaheuristic algorithms.
- The population of the metaheuristic algorithm is discretized using eight transfer functions, we select the most suitable transfer function for the proposed optimizer based on experiment results.
- Our population initialization method is applied to 5 different metaheuristic algorithms, resulting in significant performance improvement on high-dimensional feature selection problems.
- The MFO algorithm, initialized with our population initialization method, outperforms 6 currently advanced algorithms in feature selection problems.

The overall structure of our paper is as follows: Section 2 presents the background knowledge, including the WOA and its variant version, as well as the hybrid fuzzy rough set. Section 3 describes our HFRS-initialization method and the complete process of integrating our initialization method with MAs for feature selection. Section 4 conducts experimental verification. Section 5 provides a conclusion and future work. The summarized description and contribution of the HFRS-initialized MAs for feature selection in our paper are shown in Fig. 1.

2. Background

2.1. Hybrid fuzzy rough set model

Rough Set (RS) is an effective mathematical tool for processing uncertain and incomplete data. The research object of RS is decision table (information table), and no additional prior knowledge is required to analyze the data. In RS theory, knowledge is the ability to classify objects, and objects are representative of anything in the real world. The central idea of rough set theory is the approximate representation of incomplete knowledge using a known knowledge base. In other words, the known knowledge is used to approximate the unknown knowledge, provided that the classification and decision-making capabilities remain constant.

Definition 1. If U is a non-empty finite set containing several objects, then U is called a theoretical domain. $\forall X \subseteq U$ is a concept or category. And any subset of $X = \{x_1, x_2, \cdots, x_n\}$ forms the abstract knowledge in U, referred to as knowledge.

The objects of study in rough set theory are knowledge expression systems. Therefore, it is first necessary to represent these objects of study as an information system. Usually, an information system can be represented through a data table. Each row in the data table represents an object, while each column represents values of a specific attribute corresponding to the objects.

Definition 2. An information system can be written as $(U, A \cup D, V, f)$, where U is the theoretical domain, A describes the attributes of each object, which are conditional attributes; D describes the categories to which each object belongs, which are decision attributes, $V = \cup_{a \in A} V_a$ is the value domain of attributes, V_a denotes the values for attribute $a \in A$ can take, and f is the information function and can be denoted as follows:

$$f = U * (A \cup D) \rightarrow V \quad \text{Eq. (1)}$$

where f is a mapping that maps attributes of objects to values.

Attributes can be divided into categorical attribute and numeric attribute. In classical rough set model introduced in Ref. [54], crisp equivalence relations and crisp equivalence classes are used for feature selection. This model is suitable for reducing categorical attributes. Fuzzy rough set models [60,61] rely on fuzzy relations and fuzzy equivalence classes, making it more suitable for addressing the attribute reduction problem involving numeric properties.

When an information system possesses both numeric and categorical attributes, it can be referred as a hybrid information system. A hybrid information system can be denoted as $(U, A^r \cup A^c \cup D, V^r \cup V^c, f)$. Where A^r is real-valued attributes and A^c is categorical attributes. V^r and V^c are the domain of them respectively. By appropriately modifying the fuzzy rough set approach [62,75,76], it becomes possible to perform attribute reduction on hybrid information systems. The datasets used in this study are shown in Table 1. These datasets consist of both numeric attributes and categorical attributes. Therefore, we employ a fuzzy-rough model and information granulation technique for hybrid attribute reduction [62].

Definition 3. Assumed R is a fuzzy equivalence relation. We can

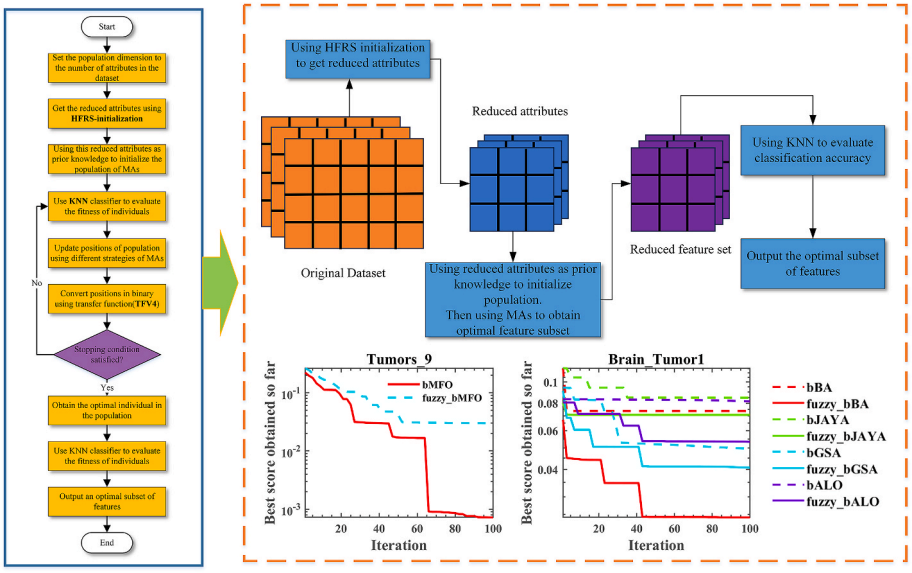


Fig. 1. The summarized description and contribution of the HFRS-initialized MAs in this study.

Table 1
The detailed descriptions of UCI datasets.

Datasets	Samples	Categories	Features	Data Type
Brain_Tumor1	90	5	5920	Categorical
Brain_Tumor2	50	4	10,367	Numerical
CNS	60	2	7130	Categorical
Colon	62	2	2000	Categorical
DLBCL	77	4	5470	Categorical
Leukemia	72	2	7131	Categorical
Leukemia1	72	5	5328	Categorical
Leukemia2	72	3	11,225	Categorical
Lung_Cancer	203	3	12,601	Numerical
Prostate_Tumor	102	2	10,509	Categorical
SRBCT	83	4	2309	Numerical
Tumors_9	60	9	5726	Categorical
Tumors_11	174	11	12,533	Categorical
Tumors_14	308	26	15,009	Categorical

represent this relation using matrix: $\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}$

Definition 4. The fuzzy equivalence class of object x_i defined by the relation R can be written as

$$[x_i]_R = \frac{r_{i1}}{x_1} + \frac{r_{i2}}{x_2} + \cdots \frac{r_{in}}{x_n} \quad \text{Eq. (2)}$$

where "+" means the union.

For $B \subseteq A$,

$$[x]_B = \cap_{a \in B} [x]_a \quad \text{Eq. (3)}$$

Where ∩ means the operations "min".

Definition 5. Assume that S_1 and S_2 are two fuzzy sets, $S_1, S_2 \subseteq U$. The inclusion $I(S_1, S_2)$ is defined as

$$I(S_1, S_2) = \frac{\|S_1 \cap S_2\|}{\|S_1\|} \quad \text{Eq. (4)}$$

Where $\|S_1\| = \frac{|S_1|}{|U|}$, $|S_1| = \sum_{x \in U} \mu_{S_1}(x)$ where μ is membership degree.

Definition 6. The variable precision lower approximations of a crisp subset X are defined as

$$\underline{R_k} X = \{x_i \mid I([x]_R, X) \geq k, x_i \in U\} \quad \text{Eq. (5)}$$

where $1 \geq k \geq 0.5$.

Definition 7. Given an information system $(U, A \cup D, V, f)$, $X_1, X_2, \ldots X_N$ are the objects partitioned by decision 1 to N, $[x_i]_B$ represents the fuzzy information granules containing x_i, which are generated from the attribute set B ⊆ A. The k-lower approximations of the decision D can be defined as

$$\underline{B_k} D = \{\underline{B_k} X_1, \underline{B_k} X_2, \ldots, \underline{B_k} X_N\} \quad \text{Eq. (6)}$$

where

$$\underline{B_k} X = \{x_i \mid I([x_i]_B, X) \geq k, x_i \in U\}, 1 \geq k \geq 0.5 \quad \text{Eq. (7)}$$

The $\underline{B_k} D$ here is also known as $POS_B^k(D)$, which is the k decision positive region of decision D as for attribute set B.

With all the above definitions, we can give out the definition of the k-dependence degree of decision D on set B.

Definition 8. k dependency of decision D on attribute set B can be defined as

$$\gamma_B^k(D) = \frac{|POS_B^k(D)|}{|U|} \quad \text{Eq. (8)}$$

If $\gamma_B^k(D) = 1$, then D is completely dependent on B. Namely that the decisions D can be fully determined by the elementary granules generated through attributes B. Given hyperparameter k, our goal is to find subset B ⊆ A. This set maximize $\gamma_B^k(D)$ while keeping $|B|$ as small as possible.

2.2. The variant of whale optimization algorithm

Whale Optimization Algorithm (WOA) [77] is proposed by Mirjalili et al. which takes clue from the hunting behavior of the humpback whale. Humpback whales employ two primary hunting strategies. The first one is sprint feeding, where the whale swiftly approaches its prey and engulfs it directly by opening its mouth. The second method involves the whale ascending in a spiral-like manner, around 15 meters below the surface, while releasing bubbles of varying sizes. These bubbles rise in unison to the surface, creating a net-like structure around the whales. When the bubbles reach the surface, the whales capture their

prey within this net. This section elaborates on the fundamental whale algorithm, which relies on the mathematical description of these two hunting behaviors.

In alignment with the two hunting phenomena, the bionic whale algorithm is also categorized into two cases: random search and bubble net attack. During the bubble net attack, since the whale's ascending trajectory resembles a spiral shape, the attack strategy includes two patterns: spiral position update and reduction of the encircling prey.

2.2.1. Spiral updating position

The mathematical model that imitates the position updates of whales using a logarithmic spiral is as follows:

$$\overrightarrow{X(t+1)} = \overrightarrow{D} \bullet e^{bl} \bullet \cos(2\pi l) + \overrightarrow{X^*(t)} \quad \text{Eq. (9)}$$

$$\overrightarrow{D} = \left|\overrightarrow{X^*(t)} - \overrightarrow{X(t)}\right| \quad \text{Eq. (10)}$$

where $\overrightarrow{X(t)}$ and $\overrightarrow{X(t+1)}$ represent the current position of the whale and the position of the whale at the next moment after updating. $\overrightarrow{X^*(t)}$ represents the best position achieved by the whale during the process of searching for prey up to that point. The constant b determines the shape of the logarithmic spiral, while l is a random number chosen from the range between -1 and 1. When l is set to 1, the whale's current position is the farthest from the optimal position, and when l is set to 0, it is the closest to the optimal position. "." represents element-wise multiplication between vectors. The mathematical model aims to simulate the way whales navigate and find their prey efficiently using these parameters to optimize their positions during the hunting process.

2.2.2. Encircling prey

$$\overrightarrow{X(t+1)} = \overrightarrow{X^*(t)} - \overrightarrow{A} \bullet \overrightarrow{D} \quad \text{Eq. (11)}$$

$$\overrightarrow{D} = \left|\overrightarrow{C} \bullet \overrightarrow{X^*(t)} - \overrightarrow{X(t)}\right| \quad \text{Eq. (12)}$$

$$\overrightarrow{A} = 2a \bullet \overrightarrow{r_1} - a \quad \text{Eq. (13)}$$

$$\overrightarrow{C} = 2\overrightarrow{r_2} \quad \text{Eq. (14)}$$

$$a = 2 - \frac{2t}{T_{max}} \quad \text{Eq. (15)}$$

The position update process of the whale involves random numbers $\overrightarrow{r_1}$, and $\overrightarrow{r_2}$, which are generated within the range of $[0, 1]$. The vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ are utilized to govern the whale's movement during the position update. The variable a holds significance in the WOA, as it impacts the vector $\overrightarrow{A}$ and indirectly controls the whale's path. The variation process of a is shown by Eq. (15), where it decreases linearly from 2 to 0 as the number of substitutions increases. T_{max} represents the predefined maximum number of iterations.

The preceding sections, 3.2.1 and 3.2.2, represent the bubble net attacks described earlier. The selection of the bubble net attack method is determined by a random number, with the probability p typically set to 0.5.

Depending on the value of p, whales employ different strategies to encircle their prey.

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X^*(t)} - \overrightarrow{A} \bullet \overrightarrow{D}, rand < p \\ \overrightarrow{D} \bullet e^{bl} \bullet \cos(2\pi l) + \overrightarrow{X^*(t)}, rand \geq p \end{cases} \quad \text{Eq. (16)}$$

2.2.3. Random search

In order to improve the searching ability of WOA and prevent it from prematurely converging to local optima, Mirjalili et al. introduced a random position update for whales when $\left|\overrightarrow{A}\right| \geq 1$. The mathematical model is shown as follows:

$$\overrightarrow{X(t+1)} = \overrightarrow{X_{rand}} - \overrightarrow{A} \bullet \overrightarrow{D_{rand}} \quad \text{Eq. (17)}$$

$$\overrightarrow{D_{rand}} = \left|\overrightarrow{C} \bullet \overrightarrow{X_{rand}} - \overrightarrow{X(t)}\right| \quad \text{Eq. (18)}$$

where $\overrightarrow{X_{rand}}$ is a randomly selected individual.

If $\left|\overrightarrow{A}\right| < 1$, the position of the WOA is still updated using the bubble spiral attack described in Section 3.2.1 and 3.2.2. Algorithm 1 gives a pseudocode of WOA.

Algorithm 1. Pseudo-code of WOA

```
Do the initialization of the parameters: T_max, Population size N, p;
Do the initialization of whale population X randomly;
Calculate the fitness for each individual X_i in whale population;
X*= the best individual
While (t < T_max)
    for each individual
        Update, A, C, p and l (A and C represent individual's component in A and C)
        if (p < 0.5)
            if(|A|<1)
                Update the position of the current individual using equation 11
            else if (|A| ≥ 1)
                Select a random individual (X_rand)
                Update the position of the current individual using equation 17
            End if
        else if (p ≥ 0.5)
            Update the position of current individual using equation 9
        end if
    end for
    Inspect each individual to see if it exceeds the range of the solution and amend it.
    Calculate each individual's fitness
    If there is a better individual, update X*
    t = t + 1
end while
return X* as the best solution
```

2.2.4. Crossover and mutation enhanced whale optimization algorithm

To enhance the exploitation capability of the WOA algorithm and reduce computation time, we employ crossover and mutation operators [78].

The mutation rate r is changed using Eq. (19), which decreases linearly from 0.9 to 0 based on the current iteration t.

$$r = 0.9 + \frac{-0.9 * (\mathrm{t} - 1)}{\mathrm{MaxIteration} - 1} \quad \text{Eq. (19)}$$

The crossover operation is applied between the result of applying the mutation operation to an individual X^{Mut} and the existing solution $X_i(t)$, following Eq. (20):

$$X_i(t+1) = Crossover\left(X^{Mut}, X_i(t)\right) \quad \text{Eq. (20)}$$

Here, t denotes the current iteration.

$Crossover(X_1, X_2)$ can get intermediate solution by switching between two input solutions x_1 and x_2 according to probability, following Eq. (21)

$$x^d = \begin{cases} x_1^d \ if \ p \geq 0.5 \\ x_2^d \ if \ p < 0.5 \end{cases} \quad \text{Eq. (21)}$$

Here d represents the d-th dimension. The pseudo-code of the variant of WOA is listed in Algorithm 2.

Algorithm 2. Pseudo-code of variant of WOA

Do the initialization of the parameters: T_{max}, Population size N, p;
Do the initialization of whale population X randomly;
Calculate the fitness for each individual X_i in whale population;
X^* = the best individual
While ($t < T_{max}$)
  for each individual
    Use equation 19 to calculate mutation rate r
    Update A, C, p and l(A and C represent individual's component in $\overrightarrow{A}$ and $\overrightarrow{C}$)
    if ($p < 0.5$)
      if ($|A|$<1)
        X^{Mut} is obtained by applying mutation operation on X^* according to mutation rate r
        Apply crossover to X^{Mut} and $X(t)$, and set the result as $X(t+1)$.
      else if ($|A| > 1$)
        Select a random individual (X_{rand})
        X^{Mut} is obtained by applying mutation operation on X_{rand} according to mutation rate r
        Apply crossover to X^{Mut} and $X(t)$, and set the result as $X(t+1)$.
      End if
    else if ($p \geq 0.5$)
      Update the current individual using equation 9
    end if
  end for
  Inspect each individual to see if it exceeds the range of the solution and amend it.
  Calculate each individual's fitness
  If there is a better solution, update X^*
  $t = t + 1$
end while
return X^*

In this paper, we call this crossover and mutation enhanced WOA as variant of WOA. It is used for the attribute reduction of hybrid fuzzy rough sets due to its stronger search capabilities and faster running speed.

2.3. K-nearest neighbor classifier

The k-nearest neighbor (KNN) [79] algorithm is a straightforward and effective learning method that does not rely on specific assumptions. It has been proven to perform exceptionally well in approximation tasks, achieving high accuracy in classification [80–82].

This method determines the new sample's class by having k nearest neighbors vote on it. The class is decided according to the minimum distance between the sample and the training points, usually calculated using the Euclidean distance. The formula to calculate the distance between two D-dimensional data points Z_1 and Z_2 is as follows:

$$Distance\ (Z_1, Z_2) = \left(\sum_{i=1}^{D} \left(z_1^i - z_2^i\right)^2 \right)^{\frac{1}{2}} \quad \text{Eq. (22)}$$

This research uses the KNN classifier to assess the fitness of features because it is fast to train, easy to implement, and achieves high classification accuracy.

3. The proposed method

3.1. Binary metaheuristic algorithms for feature selection

Original Metaheuristic Algorithms are used to handle continuous problems. Since the solution to the feature selection problem must have each component as either 0 or 1, the original MAs needsto be converted into a binary version.s Each solution can be represented as an n-dimensional vector, where the dimension of the vector is equal to the number of features. Each element of the vector can only be 0 or 1, where 0 represents that the corresponding feature is not selected, and 1 represents that the corresponding feature is selected.

Transfer functions (TFs) are a suitable transformer to transform a continuous optimization algorithm into a discrete version, rather than changing the structure of the algorithm. In general, TFs aim to generate probabilities for modifying a continuous position vector into binary form. There are eight TFs, which can be classified into two main categories, s-shaped and v-shaped, based on their shape. The figures of TFs

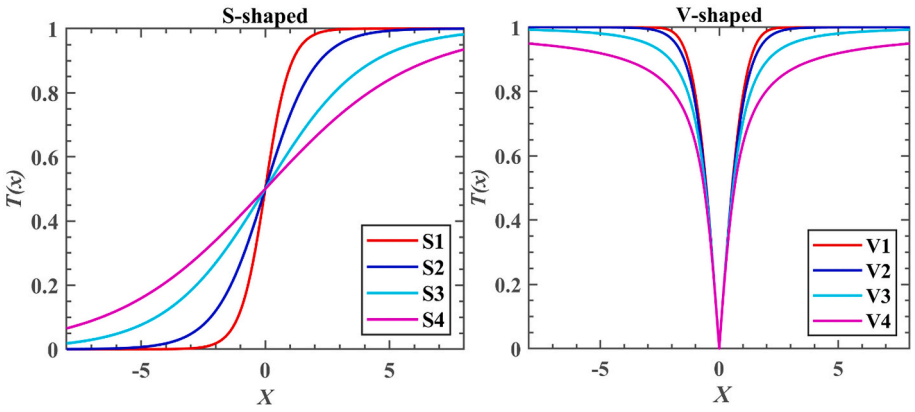


Fig. 2. S-shaped and V-shaped transfer functions.

are shown in Fig. 2. The following are the mathematical expressions of S-shaped transfer function.

$$TF_S1\left(X_i^j(t)\right)=\frac{1}{1+exp^{-2X_i^j(t)}} \quad \text{Eq. (23)}$$

$$TF_S2\left(X_i^j(t)\right)=\frac{1}{1+exp^{-X_i^j(t)}} \quad \text{Eq. (24)}$$

$$TF_S3\left(X_i^j(t)\right)=\frac{1}{1+exp^{-\left(X_i^j(t)/2\right)}} \quad \text{Eq. (25)}$$

$$TF_S4\left(X_i^j(t)\right)=\frac{1}{1+exp^{-\left(X_i^j(t)/3\right)}} \quad \text{Eq. (26)}$$

Where $X_i^j(t)$ refers to the j-th dimension of the i-th individual. Based on the calculation results, we can update the individual's position at the next moment. The update formula is as follows:

$$X_i^j(t+1)=\begin{cases}1, \text{if rand} \geq TF_S\left(X_i^j(t)\right)\\ 0, else\end{cases} \quad \text{Eq. (27)}$$

In the V-shaped family, transfer values can be calculated as follows:

$$TF_V1\left(X_i^j(t)\right)=\left|\frac{\sqrt{\pi}}{2}\int_0^{(\sqrt{\pi}/2)X_i^j(t)} e^{-t^2}dt\right| \quad \text{Eq. (28)}$$

$$TF_V2\left(X_i^j(t)\right)=\left|\tanh X_i^j(t)\right| \quad \text{Eq. (29)}$$

$$TF_V3\left(X_i^j(t)\right)=\left|\frac{X_i^j(t)}{\sqrt{1+X_i^j(t)^2}}\right| \quad \text{Eq. (30)}$$

$$TF_V4\left(X_i^j(t)\right)=\left|\frac{2}{\pi}\arctan\left(\frac{\sqrt{\pi}}{2}X_i^j(t)\right)\right| \quad \text{Eq. (31)}$$

For the V-shaped transfer function, we can use the following equation to update the individual's position at the next moment:

$$X_i^d(t+1)=\begin{cases}-X_i^d(t+1), & rand < TF_V\left(X_i^d(t+1)\right)\\ X_i^d(t+1), & else\end{cases} \quad \text{Eq. (32)}$$

By using these transfer functions and the update method, we can discretize a continuous population into one that only allows 0 and 1 to appear. original metaheuristic algorithms are transformed into binary versions. For example, we can use transfer functions to transform BA (Bat Algorithm) to bBA (binary Bat Algorithm).

3.2. HFRS-initialization

In metaheuristic algorithms, population initialization is of vital importance as it can affect the quality of the solutions. We propose a new population initialization of metaheuristic algorithms using hybrid fuzzy rough set for high-dimensional feature selection. We call this initialization method as "HFRS-initialization". Next, we will provide a detailed explanation of our HFRS initialization.

First, we normalize attributes into interval [0, 1], then we calculate relationship matrix in definition 3 for each attribute. For a specific attribute $a \in A$ and $\forall x_i, x_j \in U$, we use the following formula to calculate R_{ij}^a.

$$R_{ij}^a=\begin{cases}1-4\times\left|\mathrm{f}(x_i,a)-\mathrm{f}(x_j,a)\right|, \left|\mathrm{f}(x_i,a)-\mathrm{f}(x_j,a)\right|\leq 0.25\\ 0 \quad Otherwise.\end{cases} \quad \text{Eq. (33)}$$

This formula applies to all types of attributes, whether they are categorical or numerical attributes. If the attribute is a categorical attribute, then when two attribute values are the same, i.e., $\mathrm{f}(x_i,a)-\mathrm{f}(x_j,a)=0$, then R_{ij}^a is equal to 1. This implies that for attribute a, object x_i and object x_j belong to the same category. If they are not equal, i.e., $\left|\mathrm{f}(x_i,a)-\mathrm{f}(x_j,a)\right|>0.25$, then R_{ij}^a is equal to 0, meaning that object x_i and object x_j do not belong to the same category for attribute a. If the attributes are numerical attributes, then $\left|\mathrm{f}(x_i,a)-\mathrm{f}(x_j,a)\right|$ measures the distance between the two attribute values. The smaller the distance, the more likely they belong to the same category, and accordingly, R_{ij}^a is larger. If the distance exceeds the threshold of 0.25, it is considered that object x_i and object x_j do not belong to the same category, and R_{ij}^a is set to 0. It can be seen that the above formula unifies the calculation of the relationship matrix for categorical and numerical attributes, making it more convenient to use.

After calculating the relationship matrix, we randomly initialize the population $Y_m(m=1,2,\ldots,N)$ of the variant of WOA. Each individual represents an attribute set $B_m \subseteq A$. For these sets, we use Eq. (8) to calculate the k dependency of D on B: $\gamma_B^k(D)$. In order to balance dependencies and feature dimensions, we select the following equation as the fitness function.

$$fitness=\alpha\bullet\gamma_B^k(D)+\beta\times\frac{|N|-|R|}{|N|} \quad \text{Eq. (34)}$$

Where $|R|$ is the number of selected features, and $|N|$ is the total feature number of original dataset. We follow Maini et al. 's work [83] in which they set the values of α as 0.9 and β as 0.1. The pseudocode for calculating the fitness value of an individual (a set $B \subseteq A$) is described as follows. We name this function as *fitFun*.

Algorithm 3. Pseudo-code of *fitFun*

Input: Objects U, Objects division $[X_1, X_2, \ldots, X_N]$ according to decision 1 to N, Individual Y, Lower boundary k
Output: k dependency of D on B $\gamma_B^k(D)$

Individual Y represents a subset $B \subseteq A$
For each X in $[X_1, X_2, \ldots, X_N]$:
  For each object $x \in U$:
    calculate fuzzy equivalence class $[x]_B$ by the Eq.2 and Eq.3
    calculate the inclusion of $[x]_B$ and X by the Eq.4
    if $I([x]_B, X) \geq k$:
      x is an element of $\underline{B}_k X$
    End if
  End for
  create k-lower approximations of the decision D: $\underline{B}_k D$ by Eq.6
End for
calculate fitness of this individual by Eq.8 and Eq.34

This value returned from *fitFun* serves as the fitness value for each individual and is the optimization objective for the subsequent variant of bWOA algorithm. We use *TF_V4* to discretize the population of WOA. The best individual after executing the algorithm for the all iterations represents the final reduced attribute set. The pseudocode for HFRS-initialization is as follows:

Algorithm 4. Pseudo-code of HFRS-initialization

Do the initialization of the parameters: T_{max}, Population size N, p;
Do the initialization of whale population Y. Each $Y_m \in Y$ represents an attribute set B_m;
The dataset is scaled to range [0,1]
For each $a \in A$, calculate Relation Matrix R^a by Eq.33;
Divide all the objects X into $X_1, X_2, \ldots, X_N$ according to decision 1 to N
While ($t < T_{max}$)
  For each individual Y_m :
    $fitness_m = fitFun\ (U, [X_1, X_2, \ldots, X_N], Y_m, k)$
  End for
  Update the positions of individuals using variant of WOA (Algorithm 2)
  Transfer positions in binary using transfer function (TF_V4)
  $t = t + 1$
end While
Return Y^*

3.3. Our proposed feature selection algorithm

The goal of optimization algorithms is to reduce the fitness value calculated by fitness function. For the feature selection problem, the fitness function needs to consider two aspects: the dimension of selected features and the accuracy of the final classification. These two indicators seem somewhat contradictory, as reducing the number of selected features generally tends to lower the classification accuracy. In the face of this dilemma, there are generally two solutions. The first is to use multi-objective optimization methods [84], and the second is to integrate the two indicators into one [85].

In this paper, we use the following fitness function to strike a balance between classification accuracy and the number of selected features:

$$fitness = \alpha \bullet \gamma_R(D) + \beta \frac{|R|}{|N|} \quad \text{Eq. (35)}$$

In [86], the authors compared the influence of different values of α within the range of [0, 1] on the classification accuracy and feature length. They identified the best feature subset when the value of α was set to 0.05. In this paper, we follow the settings of their work and set α to the same value.

With the above preparation and the setting of the fitness function, we can finally begin to introduce our feature selection algorithm using HFRS initialization. In order to maintain consistency with previous AI experiments, this paper follows an established fairness standard [87–89]. As an accepted practice, ten-fold cross-validation is employed. We use a KNN classifier to evaluate the classification accuracy of the selected features. To sum up, our feature selection process consists of two parts. The first part involves attribute reduction using HFRS. The second part utilizes this well-reduced feature set as the initial prior information to carry out further feature selection using MAs.

Specifically, the population of MA is divided into two parts. One individual in the first part is set as the result of attribute reduction. The remaining individuals in first part are initialized according to the following rules: when the certain dimension of reduced attribute is 1, there is a 0.8 probability to set individual's that dimension as value 1 and a 0.2 probability of taking the value 0, whereas when the certain dimension of reduced attribute is set to 0, there is a 0.8 probability of taking the value 0 and a 0.2 probability of taking the value 1 for that dimension of that individual. On the other hand, the individuals in the second part are initialized randomly. Our proposed high-dimensional feature selection metaheuristic algorithm using hybrid fuzzy rough sets for initialization is shown in Algorithm 5.

Algorithm 5. Pseudo-code of the MAs using HFRS-initialization

The dataset is scaled to range $[0,1]$
Define the parameters of the HFRS-initialization;
Get Y^* using HFRS-initialization
The dataset is scaled to range $[-1,1]$and divided into a training set and a test set;
Define the parameters of the MAs such as population size N, The maximum number of iterations T_{max} and so on;
Set individuals' dimension equals to feature length D;
Set the location of first individual $X_1 = Y^*$
For individuals $X_i \left(i = 2,3, \ldots, \frac{N}{2}\right)$
  For dimensions $j = 1,2, \ldots, D$
    If $(Y_j^* == 1)$
      If $(rand \leq 0.8)$
        $X_{ij} = 1$
      Else
        $X_{ij} = 0$
      End If
    Else
      If $(rand \leq 0.2)$
        $X_{ij} = 1$
      Else
        $X_{ij} = 0$
      End If
    End If
  End For
End For
For individuals $X_i \left(i = \frac{N}{2} + 1, \frac{N}{2} + 2, \ldots, N\right)$
  Randomly initialize these individuals
End For
While $(t \leq T_{max})$
  Use the KNN classifier to evaluate the fitness of individuals;
  Update positions of individuals using MAs' strategy
  Transfer positions in binary;
  $t = t + 1$
End While
Return X^*

3.4. Computational complexity analysis

Our algorithm consists of two parts, and similarly, when analyzing the time complexity, we also analyze it from these two parts. The first part is attribute reduction based on hybrid fuzzy rough sets, and the second part is features selection using metaheuristic algorithms. The complexity of the first part needs to consider the following parts: calculation of the relation matrix, population initialization, evaluating fitness and population's update. The complexity of the second part needs to consider population initialization, evaluating fitness and updating population. All of the above complexities depend on population size N, max iteration number T, the number of samples U, and dimension D. The computational complexity of the attribute reduction part is as follows. The computation complexity of calculating relation matrix is $O(U^2 \times D)$. The time complexity of population initialization is $O(N)$. The time consumption of fitness evaluation is $O(T \times U^2)$, and computational complexity in updating population is $O(T \times N \times D)$. So, the total computation complexity of attribute reduction is $O(U^2 \times D + N + T \times U^2 + T \times N \times D)$. The time consumption of the feature selection part is as follows. The time complexity of initialization is $O(N \times D)$. The time consumption of fitness evaluation is $O(T \times D)$, and computational complexity in updating population is $O(T \times N \times D)$. So, the total computational complexity of feature selection is $O(N + T \times D + T \times N \times D)$. To sum up, our proposed method's computational complexity is $O(U^2 \times D + N + N \times D + T \times D + T \times U^2 + 2 \times T \times N \times D)$.

4. Results and discussion

In this study, feature selection was conducted on UCI high-dimensional gene datasets, and fitness evaluation was performed using KNN. The detailed description of the dataset is shown in Table 1. The basic settings for all metaheuristic algorithms were the same: 20 search agents were utilized, and the entire process was repeated 10 times using ten-fold cross-validation. In each run, a maximum of 100 evaluations were carried out in a consistent environment. For all methods utilizing hybrid fuzzy rough set initialization, the threshold k in Definition 6 was set to 0.85.

4.1. The proposed method with different TFs

In the following article, we use fuzzy_"algorithm name" to refer to the enhanced version of the algorithm with our HFRS-initialization. In this section, we developed a total of eight algorithms for fuzzy_bMFO, employing eight Transfer Functions (TFs) to assess their effectiveness in gene selection. Our goal was to identify the most suitable TF for fuzzy_bMFO, enabling the identification of optimal features from different datasets. The TFs used comprised of four S-shaped and four V-shaped functions. The results of our analysis, as presented in Tables A1-A4, encompass various performance metrics such as average fitness, average error rate and so on. By examining these indicators, we rank the following optimizers: fuzzy_bMFO_S1, fuzzy_bMFO_S2, fuzzy_bMFO_S3, fuzzy_bMFO_S4, fuzzy_bMFO_V1, fuzzy_bMFO_V2, fuzzy_bMFO_V3, and fuzzy_bMFO_V4.

The fitness of fuzzy_bMFO using different transfer functions is shown in Table A1. It is evident that the fitness obtained using the V-shaped function is significantly smaller than that obtained using the S-shaped function. Among all transfer functions, the fitness achieved by using the V4 function has the lowest average rank across the 14 datasets.

Table A2 displays the error rates of fuzzy_bMFO when utilizing different transfer functions. The results reveal that fuzzy_bMFO with the V3 function yields the lowest error rates when performing feature selection and classification using KNN. The V4 function follows closely with slightly higher error rates, ranking second.

Similarly, based on Table A3 and Table A4, it is observed that the fuzzy_bMFO_V4 transfer selects the smallest number of features and requires the least amount of time among all eight transfer functions.

In conclusion, fuzzy_bMFO_V4 ranks second in terms of average error rates among the eight transfer functions, while securing the first position in terms of average fitness, mean number of selected features, and average computational time. As a result, we choose the V4 function as transfer function for all the mentioned metaheuristics below.

4.2. Verification of mechanism

We will validate the effectiveness of our method from several aspects. First, we aim to demonstrate that the Rough Set (RS) method with certain searching strategy yields better results compared to other filter methods. Secondly, we will verify that using a variant of WOA on rough sets performs better than other metaheuristics. After that, we intend to confirm that the performance of feature selection improves significantly when metaheuristics are used with our initialization method compared to their original version. Finally, through experiments, we will illustrate that the performance of fuzzy_bMFO surpasses currently advanced metaheuristics, thereby demonstrating the suitability of our approach for high-dimensional feature selection problems.

4.2.1. Comparison between RS methods and other filter methods

The feature selection algorithm based on rough set is a type of filter method. In this chapter, we will experimentally validate that the performance of the Rough Set (RS) method is superior to other filter methods on UCI dataset. This superiority is manifested in two aspects: first, features selected based on RS have lower fitness compared to features selected based on filter methods. Second, the fitness of features selected using RS for population initialization followed by the execution of a metaheuristic is lower than which get from other filter methods. For comparison, we have selected MRMR [7] and ReliefF [8] as filter methods and MFO as the metaheuristic algorithm [25]. For RS-based filter, we choose variant of bWOA to find minimum reducts. For MRMR, we set the number of features it selects to be the same as the number selected using a variant of bWOA on rough sets. For ReliefF, we set $k = 3$ and $\theta = 0.6$, consistent with the settings in Ref. [51]. The experimental results are as follows:

Table 2 presents the fitness of features selected by various filter methods. Random represents random initialization, meaning no filter was applied. BWOA-variant, bWOA, bDE, and bPSO represent the respective algorithms applied on RS. MRMR and ReliefF represent the use of these two algorithms as filters to screen out features. In Table 3, bMFO, fuzzy_bMFO, bWOA_bMFO, bDE_bMFO, bPSO_bMFO represent using random method, variant of bWOA based RS method, bWOA based RS method, bDE based RS method and bPSO based method to initialize population, then utilize bMFO to conduct further feature selection. Similarly, MRMR_bMFO and reliefF_bMFO represent using MRMR and reilefF to initialize population respectively, then use bMFO to select best feature subset.

From the results in Table 2 and it is evident that the features selected using filters are indeed of higher quality compared to features initialized randomly. The fitness (Eq (35)) of feature subsets selected by bWOA-variant, bWOA, bDE, MRMR, and reliefF is reduced compared to the fitness of features initialized randomly.

Specifically, by comparing the results of bWOA-variant with MRMR and reliefF, it is evident that using features selected based on rough sets with specific metaheuristics yields better results than using conventional filters such as MRMR and reliefF. The average fitness of bWOA-variant is 0.09394, which is smaller than both MRMR with a fitness of 0.097514 and reliefF with a fitness of 0.106309. The average ranking of bWOA-variant is 2.07143, which is smaller than both MRMR with a ranking of 2.642857 and reliefF with a ranking of 2.928571.

Using rough sets for feature selection allows us to choose a superior feature subset. But can initializing the population based on this feature subset ultimately lead to better results? Based on our experimental results, the answer is affirmative. As shown in Table 3, the average fitness of fuzzy_bMFO is 1.2697E-02, which is smaller than both MRMR_bMFO with a fitness of 1.5256E-02 and reliefF_bMFO of 1.4921E-02. The average ranking of fuzzy_bMFO is 2.35714286, which is smaller than both MRMR_bMFO with a ranking of 2.571428571 and reliefF_bMFO with a ranking of 3.285714286. It's worth noting that the "RANK" row in Table 3 is generally consistent with Table 2. The higher the quality of the initialized features, the higher the quality of the features obtained after running the metaheuristic algorithm. This indicates that in the bMFO algorithm, higher-quality initialized features do indeed have a positive impact on the final feature selection results.

To sum up, the quality of the feature subset obtained through attribute reduction in RS using specific methods is higher than that obtained through other classical filter methods. Using this higher-quality subset for population initialization and subsequently running the metaheuristic results in better final solution. For more experimental results, such as error rates, feature lengths, and time consumption of different initialization methods, please refer to Table A5, A6 and A7.

Table 2
Comparison fitness of features selected by stochastic methods on RS with other filters.

	random	bWOA-variant	bWOA	bDE	bPSO	MRMR	reliefF
Brain_Tumor1	0.11458	**0.06604**	0.07878	0.080496	0.10671	0.090191	0.082211
Brain_Tumor2	0.18757	**0.0777**	0.11704	0.1373	0.20821	0.094637	0.085445
CNS	0.22705	**0.13867**	0.19665	0.20098	0.23361	0.14432	0.1837
Colon	0.23466	0.12948	**0.11284**	0.24226	0.23189	0.14417	0.17723
DLBCL	0.038399	**0.01987**	0.020828	0.035521	0.076875	0.030809	0.022411
Leukemia	0.038099	0.010374	0.011255	0.011312	0.024645	**9.82E-05**	0.003991
Leukemia1	0.036277	0.012795	0.026277	0.02222	0.037901	**0.00302**	0.009798
Leukemia2	0.050162	0.035269	0.031655	0.04829	0.051493	**0.01856**	0.023213
Lung_Cancer	0.066677	0.041314	0.043246	0.037322	0.068194	0.043081	0.039459
Prostate_Tumor	0.14657	0.12076	0.12294	0.10824	0.15495	**0.08696**	0.11401
SRBCT	0.037978	0.032194	0.032376	0.034153	0.045342	**0.00505**	0.023253
Tumors_9	0.26406	**0.22187**	0.2496	0.34845	0.32186	0.27041	0.25606
Tumors_11	0.11663	**0.08072**	0.093518	0.1114	0.108	0.10468	0.10432
Tumors_14	0.34967	**0.32807**	0.3566	0.36524	0.34923	0.32921	0.36322
AVG	0.136313	**0.09394**	0.106686	0.12737	0.144208	0.097514	0.106309
RANK-ARV	5.928571	**2.07143**	3.428571	4.785714	6.214286	2.642857	2.928571
RANK	6	**1**	4	5	7	2	3

Table 3
Comparison fitness of bMFO initialized with RS-based methods with other filter-based methods.

		bMFO	fuzzy_bMFO	bWOA_bMFO	bDE_bMFO	bPSO_bMFO	MRMR_bMFO	reliefF_bMFO
Brain_Tumor1	Avg	9.6106E-03	7.3480E-05	4.8142E-05	8.3615E-05	**4.4764E-05**	9.5312E-03	4.5608E-05
	Std	3.0021E-02	8.5768E-05	4.6269E-05	1.3662E-04	**2.6723E-05**	3.0040E-02	3.7813E-05
Brain_Tumor2	Avg	3.8102E-05	1.3504E-05	**1.1093E-05**	2.2668E-05	1.4951E-05	1.2540E-05	1.2540E-05
	Std	5.3872E-05	7.8100E-06	**3.2553E-06**	3.6381E-05	1.1912E-05	6.1007E-06	5.6612E-06
CNS	Avg	3.0860E-05	1.8235E-05	2.3145E-05	1.4729E-05	3.0159E-05	**1.2624E-05**	2.5950E-05
	Std	3.2764E-05	1.5223E-05	2.3156E-05	6.9745E-06	1.6548E-05	**6.4451E-06**	1.2393E-05
Colon	Avg	5.0000E-05	4.0000E-05	**3.7500E-05**	4.0000E-05	4.2500E-05	4.2500E-05	5.0000E-05
	Std	2.3600E-05	1.7500E-05	1.3176E-05	1.2910E-05	1.6874E-05	**1.2076E-05**	1.6667E-05
DLBCL	Avg	1.6456E-05	1.2799E-05	1.5542E-05	1.3714E-05	1.8285E-05	**1.2799E-05**	1.4628E-05
	Std	5.7822E-06	**4.7211E-06**	8.6733E-06	6.4647E-06	4.3098E-06	4.7211E-06	4.7211E-06
Leukemia	Avg	1.3326E-05	1.1222E-05	9.8190E-06	9.8190E-06	1.1923E-05	**7.0136E-06**	9.8190E-06
	Std	3.9812E-06	3.6218E-06	3.6218E-06	3.6218E-06	5.7741E-06	**0.0000E + 00**	3.6218E-06
Leukemia1	Avg	2.4404E-05	2.0650E-05	**1.9711E-05**	2.6281E-05	3.7545E-05	2.0650E-05	2.0650E-05
	std	1.4808E-05	**7.4039E-06**	8.2185E-06	2.2035E-05	5.6317E-05	1.0656E-05	7.4039E-06
Leukemia2	avg	1.2027E-05	1.0690E-05	1.0690E-05	1.2918E-05	1.2918E-05	**8.0178E-06**	8.4633E-06
	std	5.1648E-06	4.3033E-06	3.7562E-06	3.9002E-06	5.7313E-06	2.8172E-06	**1.4086E-06**
Lung_Cancer	avg	4.9230E-03	8.6905E-05	1.1706E-04	**5.2381E-05**	4.6448E-03	1.3175E-04	1.2817E-04
	std	1.5000E-02	7.2413E-05	1.0513E-04	**3.5285E-05**	1.4383E-02	1.7670E-04	1.5580E-04
Prostate_Tumor	avg	3.7111E-05	2.2838E-05	3.7111E-05	3.7111E-05	4.1869E-05	**1.2846E-05**	3.2829E-05
	std	3.5449E-05	1.7052E-05	5.5612E-05	5.3632E-05	4.6929E-05	**6.3636E-06**	4.5050E-05
SRBCT	avg	6.2825E-05	5.8492E-05	7.5823E-05	8.0156E-05	6.9324E-05	**5.6326E-05**	6.2825E-05
	std	1.2297E-05	1.7835E-05	2.7498E-05	2.2950E-05	1.9908E-05	1.5147E-05	2.9686E-05
Tumors_9	avg	2.9972E-02	**7.2564E-04**	3.2863E-02	1.4452E-03	2.7801E-02	1.9601E-02	1.4104E-02
	std	6.2020E-02	**6.6792E-04**	6.9729E-02	1.1499E-03	5.7350E-02	5.9906E-02	4.2906E-02
Tumors_11	avg	5.8802E-03	**1.4889E-03**	6.6407E-03	1.2233E-02	1.1541E-02	6.9295E-03	1.5886E-03
	std	1.4531E-02	**9.0211E-04**	1.7653E-02	2.2813E-02	2.2015E-02	1.7336E-02	1.0555E-03
Tumors_14	avg	1.9195E-01	**1.7518E-01**	1.8835E-01	1.9461E-01	1.8593E-01	1.7720E-01	1.9279E-01
	std	6.0664E-02	**7.4950E-02**	7.1483E-02	5.0710E-02	6.5659E-02	4.6727E-02	6.6926E-02
AVG		1.7330E-02	**1.2697E-02**	1.6304E-02	1.4906E-02	1.6446E-02	1.5256E-02	1.4921E-02
RANK-ARV		5.57142857	**2.35714286**	3.428571429	4.285714286	5.285714286	2.571428571	3.285714286
RANK		7	1	4	5	6	2	3

Table 4
The parameters of MAs.

Method	Reference	Common parameters	Parameters
DE	[90]	Dimension is same with feature num, Population size equals 20	cr = 0.1; f = 0.5;
PSO	[91]	Dimension is same with feature num, Population size equals 20	$w = 2$; $wMax = 0.9$; $wMin = 0.4$; $c1 = 2; c2 = 2$; $Vmax = 6$
WOA	[77]	Dimension is same with feature num, Population size equals 20	$b = 1$

4.2.2. Comparison between variant of WOA and other metaheuristics on RS

In this chapter, we aim to demonstrate that the variant of bWOA outperforms other metaheuristic algorithms in RS attribute reduction. The algorithms we will compare, along with their parameter settings, are listed in Table 4.

WOA algorithm has various improved versions. In this work, we used the Cross-Over and Mutation Enhanced WOA because it offers stronger search capabilities compared to the original version and lower time complexity compared to more advanced versions. More advanced version such as modified WOA with population reduction introduced by Chakraborty et al. [92], which initializes the population in each iteration, increasing the algorithm's complexity. Xing et al. proposed quasi-oppositional learning and Gaussian barebone to WOA [93]. Such complex search strategies further increase the algorithm's complexity. Through the experiments in **chapter 4.2.4**, we can observe that the features selected using the variant of bWOA in this paper already achieve high quality. The fitness is lower than six currently advanced algorithms, so there is no need to introduce complex strategies that would burden computation. In this chapter, we only compare the features

Table 5
Comparison RS fitness of metaheuristics and other filters.

	random	bWOA-variant	bWOA	bDE	bPSO	MRMR	reliefF
Brain_Tumor1	0.9501	**0.99971**	0.99951	0.99002	0.95115	0.92971	0.12998
Brain_Tumor2	0.95033	**0.99986**	0.99978	0.99903	0.95054	0.67586	0.13599
CNS	0.95009	**0.99978**	0.99976	0.99314	0.95068	0.80478	0.099986
Colon	0.9483	**0.9997**	0.9996	0.99245	0.9519	0.94164	0.09995
DLBCL	0.94946	0.99967	**0.99969**	0.99896	0.9512	0.98798	0.55583
Leukemia	0.95041	0.9998	**0.99982**	0.9967	0.95082	0.9373	0.89996
Leukemia1	0.94995	**0.99977**	**0.99977**	0.99874	0.95181	0.81227	0.43748
Leukemia2	0.94901	0.99986	0.99988	0.99335	0.95129	0.81236	0.099991
Lung_Cancer	0.94983	**0.99981**	0.9997	0.99548	0.95064	0.95991	0.33497
Prostate_Tumor	0.9501	**0.99964**	0.99943	0.99726	0.95106	0.92905	0.15293
SRBCT	0.9487	**0.99944**	**0.99944**	0.99476	0.9529	0.93438	0.38188
Tumors_9	0.94921	0.99976	**0.99977**	0.9926	0.95068	0.86476	0.11498
Tumors_11	0.95004	**0.99975**	0.99971	0.99363	0.95063	0.98941	0.11551
Tumors_14	0.95022	**0.99516**	0.98818	0.99061	0.95079	0.96009	0.099993
AVG	0.949696	**0.99941**	0.99886	0.994766	0.951149	0.895679	0.261388
RANK-ARV	5.285714	**1.28571**	1.642857	2.928571	4.285714	5.428571	7
RANK	5	**1**	2	3	4	6	7

selected by the variant of WOA based on RS with those selected by the original WOA and algorithms with similar computation complexity like Differential Evolution (DE) and Particle Swarm Optimization (PSO).

The fitness of selected features using metaheuristics based on RS are presented in Table 5. Here, "fitness" refers to the value calculated by Eq. (34), which is a comprehensive measure of the selected feature's dependency and feature length. The optimization objective of metaheuristics is to maximize this value, hence it reflects the search capabilities of these meta-algorithms.

From Table 5, we can draw the following conclusions. Firstly, it is undeniable that using metaheuristics to perform attribute reduction on hybrid fuzzy rough sets results in features with higher RS fitness compared to randomly selected features or those selected using other filters. (In the context of RS, the higher fitness the better feature subset), as it is the optimization goal of these metaheuristics. We can see that the rankings of bWOA-variant, bWOA, bDE, and bPSO are higher than those of random, MRMR, and reliefF. Secondly, the quality of features obtained through WOA is superior to the other compared metaheuristics. The fitness of features obtained by bWOA-variant and bWOA is higher than that of bDE and bPSO. Lastly, from the data in the table, it's evident that the fitness of bWOA-variant is higher than that of bWOA, indicating that the mutation and crossover strategies we used indeed enhance the search capabilities of bWOA.

Table 5 displays the RS fitness of features found by various metaheuristics. How does the feature subset reduced based on RS fitness perform in terms of classification performance using KNN as the classifier? The rankings in Table 2 from the previous chapter reflect this performance.

In Table 2, we can see that the rankings of bWOA-variant and bWOA are higher than bDE and bPSO. Meanwhile, the ranking of bWOA-variant is higher than bWOA, which aligns perfectly with the results from Table 5. Similarly, by examining the data in Table 3 from the previous chapter, we can observe that using bWOA-variant for RS based initialization can lead to better final solution than using bWOA, bDE, and bPSO. For more experimental results regarding RS dependency and feature selected rate, please refer to Table A8 and Table A9.

In summary, this chapter demonstrates that the variant of bWOA is a superior rough set attribute reduction method compared to other metaheuristics, while the previous chapter establishes that rough set methods excel as filter methods. Therefore, these findings collectively indicate that our HFRS-initialization is a superior filter-based feature selection method.

Finally, let's take a more intuitive look at the difference between the population fitness after using HFRS initialization and the population fitness after random initialization (the values in the columns "random" and "bWOA-variant" in Table 2). As shown in Fig. 3, our initialization method results in higher-quality features compared to randomly selected features across various datasets. In some datasets, the fitness has decreased by more than 50%. It is worth noting that we did not incorporate KNN classification performance as a metric (Eq. (35)) to optimize during the initialization process. We only use fitness based on hybrid fuzzy rough set (Eq. (34)). If we were to include the metric in Eq. (35) as an optimization target, our initialization-phase fitness could be even lower. We chose not to include it to prevent subsequent wrapper-based metaheuristic methods from getting stuck in local optima [94–96]. We aim to make the optimization objectives in the initialization phase and wrapper-based feature selection phase completely different, so that the initialization phase can provide inspiration to the subsequent algorithms from a different perspective.

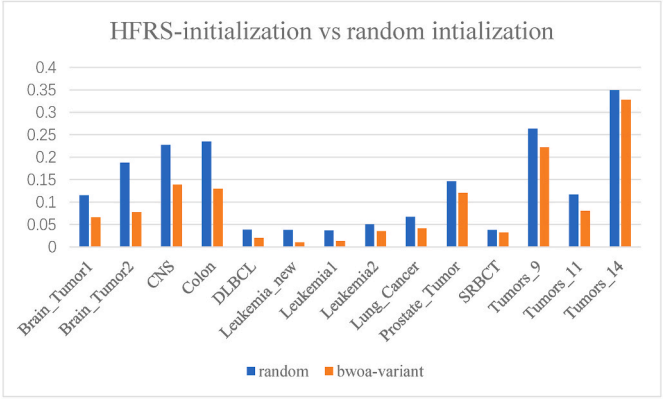


Fig. 3. Comparison fitness of HFRS with random initialization.

4.2.3. *Comparison between fuzzy versions of metaheuristics and their original ones*

To evaluate the efficiency of our initialization method, we selected a total of five algorithms and fourteen UCI high dimensional gene datasets for verification. The details of these algorithms are shown in Table 6.

To verify that our initialization method can effectively improve performance of high-dimensional feature selection problem, we will evaluate our algorithm from four perspectives: fitness, error, feature length, and time consumption. We employed the Wilcoxon signed-rank test to assess the significant differences in our modification. If the significance value (p value) is less than 0.05, we say the difference between two data is significant.

First, we evaluate the improvement in fitness after applying HFRS-initialization. Fitness is the most important evaluation metric as it considers both the accuracy of classification and the feature length after feature selection. The improvement in fitness is the strongest evidence of the effectiveness of the algorithm enhancement. The smaller the fitness, the higher the accuracy of classification and the fewer selected feature dimensions. For the bMFO algorithm, as we can observe in Table 7, on all 14 high-dimensional feature datasets, the fuzzy_bMFO algorithm shows improvements in fitness, and the p-value from the Wilcoxon test across all datasets is less than 0.05, indicating that the improvement is significant considering all 14 datasets.

This improvement shows a similar trend in other algorithms as well. As shown in Table 8, regarding fitness alone, in the bBA and bGSA, their fuzzy version shows improvement on all 14 datasets. As for bJAYA and its fuzzy version, the improved algorithm has slightly higher fitness on the Tumor_11 dataset but lower fitness on other datasets compared to the original version. As for the bALO algorithm, its fuzzy version shows improvement in twelve out of the fourteen datasets, with slightly higher fitness in Prostate_Tumor and Tumors_14. On average, the algorithms bBA, bJAYA, bGSA, and bALO show a decrease in fitness after using HFRS-initialization. Additionally, the p-values from the Wilcoxon test are all less than 0.05, indicating that this decrease is significant. Therefore, we can say that HFRS-initialization significantly reduces the fitness of the five algorithms above, resulting in higher accuracy in selecting high-dimensional features and lower feature dimensions.

We compare the convergence values between fuzzy_bMFO and bMFO in Fig. 4. It can be observed that with our initialization method, bMFO exhibits a decrease in convergence value compared to the original

Table 6
The parameters of MAs in this paper.

Method	Reference	Common parameters	Parameters
BA	[21]	Dimension is same with feature num, Population size equals 20	$a = 0.5$; $r = 0.5$;
JAYA	[22]	Dimension is same with feature num, Population size equals 20	*None*
GSA	[23]	Dimension is same with feature num, Population size equals 20	$ch = 20$; $R_{power} = 1$
ALO	[24]	Dimension is same with feature num, Population size equals 20	*None*
MFO	[25]	Dimension is same with feature num, Population size equals 20	$a \in [-1, -2]$; $t = [-1, 1]$ $b = 1$;

Table 7
The comparison of fitness, error, length and time cost of bMFO with its fuzzy version.

	fitness		error		length		time	
	bMFO	fuzzy_bMFO	bMFO	fuzzy_bMFO	bMFO	fuzzy_bMFO	bMFO	fuzzy_bMFO
Brain_Tumor1	9.6106E-03	**7.3480E-05**	1.0000E-02	**0.0000E + 00**	13.1	**8.7**	7.6369E+00	**7.6087E + 00**
	3.0021E-02	**8.5768E-05**	3.1623E-02	**0.0000E + 00**	22.8641	**10.1549**	**7.0883E-01**	1.2459E+00
Brain_Tumor2	3.8102E-05	**1.3504E-05**	**0.0000E + 00**	**0.0000E + 00**	7.9	**2.8**	1.3273E+01	**1.2329E + 01**
	5.3872E-05	**7.8100E-06**	**0.0000E + 00**	**0.0000E + 00**	11.1699	**1.6193**	**1.8161E + 00**	2.0881E+00
CNS	3.0860E-05	**1.8235E-05**	**0.0000E + 00**	**0.0000E + 00**	4.4	**2.6**	7.8539E+00	**6.5909E + 00**
	3.2764E-05	**1.5223E-05**	**0.0000E + 00**	**0.0000E + 00**	4.6714	**2.1705**	1.6685E+00	**2.9042E-01**
Colon	5.0000E-05	**4.0000E-05**	**0.0000E + 00**	**0.0000E + 00**	2	**1.6**	3.4654E+00	**2.3174E + 00**
	2.3600E-05	**1.7500E-05**	**0.0000E + 00**	**0.0000E + 00**	0.94281	**0.69921**	1.5889E+00	**2.3124E-01**
DLBCL	1.6456E-05	**1.2799E-05**	**0.0000E + 00**	**0.0000E + 00**	1.8	**1.4**	6.1378E+00	**5.8781E + 00**
	5.7822E-06	**4.7211E-06**	**0.0000E + 00**	**0.0000E + 00**	0.63246	**0.5164**	**2.4962E-01**	2.5312E-01
Leukemia	1.3326E-05	**1.1222E-05**	**0.0000E + 00**	**0.0000E + 00**	1.9	**1.6**	8.5587E+00	**7.6801E + 00**
	3.9812E-06	**3.6218E-06**	**0.0000E + 00**	**0.0000E + 00**	0.56765	**0.5164**	1.4091E+00	**6.4290E-01**
Leukemia1	2.4404E-05	**2.0650E-05**	**0.0000E + 00**	**0.0000E + 00**	2.6	**2.2**	6.2098E+00	**5.6155E + 00**
	1.4808E-05	**7.4039E-06**	**0.0000E + 00**	**0.0000E + 00**	1.5776	**0.78881**	9.8002E-01	**1.5306E-01**
Leukemia2	1.2027E-05	**1.0690E-05**	**0.0000E + 00**	**0.0000E + 00**	2.7	**2.4**	**1.0996E + 01**	1.1445E+01
	5.1648E-06	**4.3033E-06**	**0.0000E + 00**	**0.0000E + 00**	1.1595	**0.96609**	**4.2554E-01**	6.8946E-01
Lung_Cancer	4.9230E-03	**8.6905E-05**	5.0000E-03	**0.0000E + 00**	43.6	**21.9**	2.5833E+01	**2.2169E + 01**
	1.5000E-02	**7.2413E-05**	1.5811E-02	**0.0000E + 00**	42.7348	**18.248**	3.7404E+00	**9.5745E-01**
Prostate_Tumor	3.7111E-05	**2.2838E-05**	**0.0000E + 00**	**0.0000E + 00**	7.8	**4.8**	1.2463E+01	**1.2346E + 01**
	3.5449E-05	**1.7052E-05**	**0.0000E + 00**	**0.0000E + 00**	7.4506	**3.5839**	**4.3900E-01**	4.7535E-01
SRBCT	6.2825E-05	**5.8492E-05**	**0.0000E + 00**	**0.0000E + 00**	2.9	**2.7**	3.3113E+00	**3.2430E + 00**
	1.2297E-05	**1.7835E-05**	**0.0000E + 00**	**0.0000E + 00**	**0.56765**	0.82327	1.1416E-01	**5.4758E-02**
Tumors_9	2.9972E-02	**7.2564E-04**	3.0952E-02	**0.0000E + 00**	**65**	83.1	6.9454E+00	**6.2052E + 00**
	6.2020E-02	**6.6792E-04**	6.5494E-02	**0.0000E + 00**	**52.0683**	76.4903	7.0491E-01	**1.3238E-01**
Tumors_11	5.8802E-03	**1.4889E-03**	4.7619E-03	**0.0000E + 00**	**340**	373.2	2.4119E+01	**2.2549E + 01**
	1.4531E-02	**9.0211E-04**	1.5058E-02	**0.0000E + 00**	**172.2085**	226.1233	1.9696E+00	**1.1377E + 00**
Tumors_14	1.9195E-01	**1.7518E-01**	1.9469E-01	**1.7872E-01**	2097.1	**1621.3**	5.4686E+01	**5.1435E + 01**
	6.0664E-02	7.4950E-02	**6.2629E-02**	7.7273E-02	908.9571	**844.9396**	5.4963E+00	**3.1833E + 00**
AVG	1.7330E-02	**1.2697E-02**	1.7529E-02	**1.2766E-02**	185.2000	**152.1643**	1.3678E+01	**1.2672E + 01**
P-VALUE	**0.00012207**		0.0625		0.075317		**0.0012207**	

Table 8
The comparison fitness values of MAs with their fuzzy version.

		bBA	fuzzy_bBA	bJAYA	fuzzy_bJAYA	bGSA	fuzzy_bGSA	bALO	fuzzy_bALO
Brain_Tumor1	Avg	7.3721E-02	**2.4213E-02**	8.4806E-02	**7.0762E-02**	4.9709E-02	**4.0762E-02**	8.1555E-02	**5.3384E-02**
	Std	**6.8688E-02**	4.1487E-02	8.5207E-02	**6.8830E-02**	4.7501E-02	**4.7332E-02**	6.7838E-02	**5.1907E-02**
Brain_Tumor2	Avg	7.1327E-02	**2.3740E-02**	8.0688E-02	**6.1902E-02**	7.6640E-02	**5.4527E-02**	1.0227E-01	**8.0342E-02**
	Std	9.3449E-02	**6.0126E-02**	1.8020E-01	**8.1812E-02**	1.1826E-01	**8.9778E-02**	1.2933E-01	**9.2456E-02**
CNS	Avg	6.3481E-02	**2.2087E-02**	1.1895E-01	**9.7781E-02**	8.4202E-02	**4.2256E-02**	8.6301E-02	**7.8073E-02**
	Std	7.6135E-02	**4.9824E-02**	**8.1551E-02**	1.3842E-01	**8.1683E-02**	8.6348E-02	1.1081E-01	**9.9081E-02**
Colon	Avg	1.3391E-01	**4.8073E-02**	1.6583E-01	**1.1553E-01**	1.3915E-01	**6.9846E-02**	1.6132E-01	**5.1266E-02**
	Std	1.4819E-01	**7.3770E-02**	1.5742E-01	**1.0200E-01**	1.2355E-01	**8.0920E-02**	1.1653E-01	**6.8532E-02**
DLBCL	Avg	2.5934E-02	**4.4825E-03**	3.6943E-02	**2.1167E-02**	3.1572E-02	**6.9373E-03**	2.2388E-02	**2.0741E-02**
	Std	3.7534E-02	**1.5791E-03**	4.2826E-02	**3.8286E-02**	3.7474E-02	**4.6825E-03**	**4.4164E-04**	3.8050E-02
Leukemia	Avg	1.3944E-02	**5.8290E-03**	2.3354E-02	**1.2553E-02**	2.0257E-02	**8.5994E-03**	2.2557E-02	**1.1095E-02**
	Std	**5.1641E-04**	1.9037E-03	**1.1201E-04**	5.3822E-03	**3.1287E-04**	5.1784E-03	**1.7914E-04**	4.1837E-03
Leukemia1	Avg	1.4097E-02	**5.5162E-03**	2.3341E-02	**1.2558E-02**	1.9755E-02	**1.0304E-02**	2.2255E-02	**1.1011E-02**
	Std	**6.1954E-04**	9.7531E-04	**1.9260E-04**	5.3928E-03	**1.6058E-04**	2.3097E-03	**3.4899E-04**	4.0822E-03
Leukemia2	Avg	3.9132E-02	**5.0548E-03**	2.3598E-02	**2.3518E-02**	3.2940E-02	**1.1617E-02**	2.3181E-02	**1.1262E-02**
	Std	5.3803E-02	**7.0395E-04**	**3.7271E-04**	3.7627E-02	3.7522E-02	**3.3375E-03**	**2.1224E-04**	3.5745E-03
Lung_Cancer	Avg	3.3327E-02	**2.0709E-02**	4.6963E-02	**3.5133E-02**	4.4352E-02	**3.1014E-02**	5.1386E-02	**3.5155E-02**
	Std	**2.3308E-02**	2.3616E-02	3.3238E-02	**2.9163E-02**	3.2106E-02	**3.1731E-02**	3.2993E-02	**2.4970E-02**
Prostate_Tumor	Avg	7.2333E-02	**2.6361E-02**	8.0027E-02	**6.0499E-02**	6.8041E-02	**4.3022E-02**	**5.0953E-02**	6.1450E-02
	Std	9.2006E-02	**4.2145E-02**	6.5988E-02	**6.3048E-02**	**4.9271E-02**	6.3500E-02	**4.4591E-02**	6.8809E-02
SRBCT	Avg	1.3761E-02	**5.3510E-03**	3.4687E-02	**1.5312E-02**	1.6785E-02	**9.7465E-03**	2.1157E-02	**1.0812E-02**
	Std	2.0844E-03	**2.0025E-03**	3.7651E-02	**6.8695E-03**	**6.9368E-04**	2.4033E-03	**4.0788E-04**	3.3654E-03
Tumors_9	Avg	1.3421E-01	**4.8655E-02**	2.0384E-01	**1.6266E-01**	1.4316E-01	**9.2129E-02**	1.6272E-01	**1.5242E-01**
	Std	**8.7092E-02**	9.1352E-02	**1.1665E-01**	1.1928E-01	1.1865E-01	**8.0139E-02**	1.5377E-01	**8.4566E-02**
Tumors_11	Avg	5.8503E-02	**1.9042E-02**	**7.6775E-02**	8.0609E-02	4.9158E-02	**4.3646E-02**	7.3315E-02	**5.0866E-02**
	Std	5.9478E-02	**2.3030E-02**	**4.7073E-02**	5.8761E-02	3.8899E-02	**3.0087E-02**	**1.8345E-02**	5.4042E-02
Tumors_14	Avg	2.8804E-01	**2.6866E-01**	3.3238E-01	**3.2843E-01**	2.5945E-01	**2.5446E-01**	**2.7394E-01**	2.8793E-01
	Std	6.3360E-02	**3.0029E-02**	**7.1195E-02**	8.5959E-02	7.7624E-02	**6.2582E-02**	**5.5869E-02**	6.6811E-02
AVG		7.3980E-02	**3.7698E-02**	9.5156E-02	**7.8458E-02**	7.3941E-02	**5.1348E-02**	8.2521E-02	**6.5415E-02**
P-VALUE		**0.00012207**		**0.00036621**		**0.00012207**		**0.013428**	

version across various datasets. As for other algorithms, their fuzzy versions generally have lower convergence values than the original versions in fourteen datasets, as detailed in Fig. 5.

Second, we evaluate the improvement in error rate after applying HFRS-initialization. As shown in Table 7, compared to bMFO, fuzzy_bMFO effectively reduces the error rates on all datasets (those with error rates of 0 when using bMFO for feature selection and then applying knn classification remain at 0 in the improved version of the algorithm).

Table 9
The comparison error rate of MAs with their fuzzy version.

		bBA	fuzzy_bBA	bJAYA	fuzzy_bJAYA	bGSA	fuzzy_bGSA	bALO	fuzzy_bALO
Brain_Tumor1	Avg	6.3333E-02	**2.1111E-02**	6.4444E-02	**6.2222E-02**	3.1111E-02	**3.1111E-02**	6.2222E-02	**4.3333E-02**
	Std	7.2208E-02	**4.4583E-02**	8.9779E-02	**7.1473E-02**	**5.0185E-02**	**5.0185E-02**	7.1473E-02	**5.6035E-02**
Brain_Tumor2	Avg	5.9286E-02	**2.0000E-02**	6.0000E-02	**5.4286E-02**	5.8333E-02	**4.5000E-02**	8.3333E-02	**7.0952E-02**
	Std	9.8748E-02	**6.3246E-02**	1.8974E-01	**8.8781E-02**	1.2454E-01	**9.5598E-02**	1.3608E-01	**9.3002E-02**
CNS	Avg	5.0000E-02	**1.6667E-02**	1.0000E-01	**8.3333E-02**	6.6667E-02	**2.8571E-02**	6.6667E-02	**6.1905E-02**
	Std	8.0508E-02	**5.2705E-02**	**8.6066E-02**	1.4164E-01	**8.6066E-02**	9.0351E-02	1.1653E-01	**1.0481E-01**
Colon	Avg	1.2619E-01	**4.7619E-02**	1.5000E-01	**1.1190E-01**	1.2857E-01	**6.4286E-02**	1.4762E-01	**4.5238E-02**
	Std	1.5595E-01	**7.6947E-02**	1.6574E-01	**1.1057E-01**	1.3051E-01	**8.3277E-02**	1.2285E-01	**7.3128E-02**
DLBCL	Avg	1.2500E-02	**0.0000E + 00**	1.4286E-02	**1.2500E-02**	1.2500E-02	**0.0000E + 00**	**0.0000E + 00**	1.2500E-02
	Std	3.9528E-02	**0.0000E + 00**	4.5175E-02	**3.9528E-02**	3.9528E-02	**0.0000E + 00**	**0.0000E + 00**	3.9528E-02
Leukemia	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Leukemia1	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Leukemia2	Avg	2.6786E-02	**0.0000E + 00**	**0.0000E + 00**	1.2500E-02	1.2500E-02	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	5.6626E-02	**0.0000E + 00**	**0.0000E + 00**	3.9528E-02	3.9528E-02	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Lung_Cancer	Avg	1.9286E-02	**1.4787E-02**	**2.4524E-02**	**2.4524E-02**	2.3831E-02	**1.8853E-02**	2.9524E-02	**1.9524E-02**
	Std	2.4907E-02	**2.3848E-02**	**3.4994E-02**	**3.4994E-02**	3.3706E-02	**3.2451E-02**	3.4671E-02	**2.5218E-02**
Prostate_Tumor	avg	5.9091E-02	**2.0000E-02**	5.9091E-02	**4.0000E-02**	4.9091E-02	**3.0000E-02**	**2.9091E-02**	5.0000E-02
	std	9.6233E-02	**4.2164E-02**	6.9400E-02	**6.9921E-02**	**5.1817E-02**	6.7495E-02	**4.6906E-02**	7.0711E-02
SRBCT	avg	0.0000E+00	**0.0000E + 00**	1.2500E-02	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	std	0.0000E+00	**0.0000E + 00**	3.9528E-02	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Tumors_9	avg	1.2274E-01	**4.1071E-02**	1.8952E-01	**1.5036E-01**	1.2917E-01	**7.8571E-02**	1.4714E-01	**1.4123E-01**
	std	**9.1938E-02**	9.4510E-02	1.2276E-01	**1.2156E-01**	1.2492E-01	**8.3277E-02**	1.6191E-01	**8.7860E-02**
Tumors_11	avg	4.5139E-02	**1.1438E-02**	**5.5442E-02**	6.1278E-02	2.8466E-02	**2.7659E-02**	5.2414E-02	**3.4692E-02**
	std	6.3246E-02	**2.4126E-02**	**4.9747E-02**	6.0206E-02	4.0899E-02	**2.9357E-02**	**1.9320E-02**	5.4574E-02
Tumors_14	avg	2.8298E-01	**2.6665E-01**	3.2454E-01	**3.2038E-01**	2.4866E-01	**2.4526E-01**	**2.6296E-01**	2.7852E-01
	std	6.7540E-02	**3.3240E-02**	**7.4762E-02**	9.0595E-02	8.1858E-02	**6.6220E-02**	**5.9018E-02**	7.0540E-02
AVG		6.1952E-02	**3.2810E-02**	7.5311E-02	**6.6663E-02**	5.6350E-02	**4.0665E-02**	6.2927E-02	**5.4135E-02**
P-VALUE		**0.00097656**		0.056641		**0.0019531**		0.49219	

Fuzzy_bBA, fuzzy_bJAYA, fuzzy_bGSA and fuzzy_bALO, compared to their respective original algorithms, also effectively reduce the classification error rates on most datasets, and the average error rates of all datasets have been reduced (see Table 9).

Third, we evaluate the improvement in the number of selected features after applying HFRS-initialization. As shown in Table 7 (column 'length'), for the bMFO algorithm, the number of selected features has been reduced using our initialization method in all datasets except for Tumors_9 and Tumors_11. For the bBA, bJAYA, bALO bGSA, their fuzzy version reduces feature number in all datasets. Performing a Wilcoxon test on feature number selected by the original versions of these four algorithms and their fuzzy versions in the 14 datasets yields a *p*-value less than 0.05, meaning that the reduction in feature number is significant. The details of the four algorithms and their fuzzy versions' selected feature number can be found in Table 10.

Finally, we analyzed the differences in execution time between using HFRS-initialization and not using it in the algorithms. Since for each dataset, executing the preprocessing step results in reduced features that can be used for all subsequent MAs, we only compared the time consumed by the MAs before and after using reduced features in the article, without separately calculating the time spent on HFRS-initialization.

According to Tables 7 and 11, considering all fourteen datasets, the fuzzy_bJAYA algorithm shows a slightly increased time consumption compared to bJAYA. On the other hand, the fuzzy versions of bMFO, bBA, bGSA, and bALO exhibit a decreased average time consumption compared to their original versions. By calculating the *p*-values for the time consumption on the fourteen datasets between the five original algorithms and their enhanced versions using HFRS initialization, we observe that fuzzy_bMFO, fuzzy_bBA, fuzzy_bGSA and fuzzy_bALO significantly reduce the time consumption compared to their original versions. All the compared algorithms' time consumption on all datasets is fully presented in Fig. 6.

4.2.4. Comparison between fuzzy_bMFO and currently advanced MAs

In this chapter we compared the fuzzy_bMFO algorithm with other currently advanced MAs: chaotic mutative MFO (bCLSGMFO) [97], adaptive weights levy-assisted SSA (bWLSSA) [98], weighted DE (bWDE) [99], PSO with an aging leader and challengers (bALCPSO) [100], dispersed foraging SMA (bDFSMA) [101] and generalized oppositional teaching-learning based optimization (bGOTLBO) [102]. According to Ref. [13], the average fitness of these algorithms for feature selection tasks on all high-dimensional datasets is as follows: 0.01368, 0.02129, 0.02535, 0.05366, 0.01808, 0.0213. Our fuzzy-bMFO's average fitness across all dataset is 0.012761 which is the lowest among all these currently advanced MAs, ranking first. The corresponding fitness value of the original bMFO is 0.017330, which is higher than bCLSGMFO. Among those advanced algorithms, it can only be ranked second. The superior performance of fuzzy_bMFO demonstrates that MAs enhanced with HFRS-initialization can perform exceptionally well on high-dimensional feature selection tasks.

5. Conclusions and future works

We propose a novel population initialization method called "HFRS-initialization." It relies on hybrid fuzzy rough set and employs a variant of bWOA for attribute reduction. In order to use discretized metaheuristic algorithms, we conducted experiments to test various transfer functions and ultimately selected TF_V4 to discretize the continuous positions.

To validate the effectiveness of our initialization method, we conducted a total of four experiments. Firstly, we verified that the use of a specific metaheuristic based on hybrid fuzzy rough set (also a type of filter method) resulted in higher feature quality compared to other filter methods. Then, we confirmed that the effectiveness of attribute reduction on rough sets using variant-bWOA is superior to other metaheuristics of similar complexity. Since variant-bWOA has already demonstrated excellent performance, we no longer resort to more complex algorithms for rough set attribute reduction. Moreover, through the experiments in these two sections, it is evident that a high-quality initial population has a positive impact on the final algorithm results. After that, we evaluated the performance improvement achieved

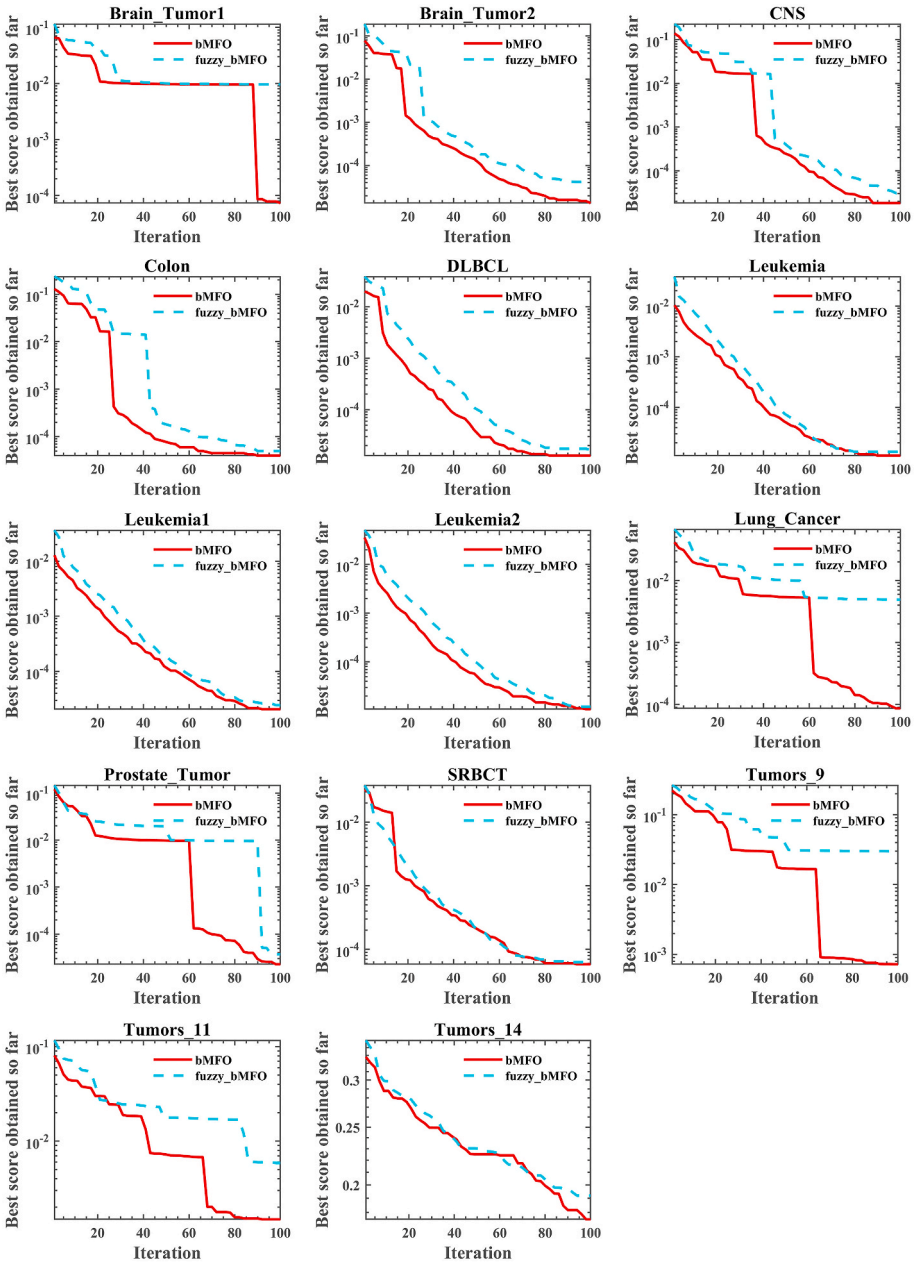


Fig. 4. The convergence curve of the bMFO and fuzzy_bMFO.

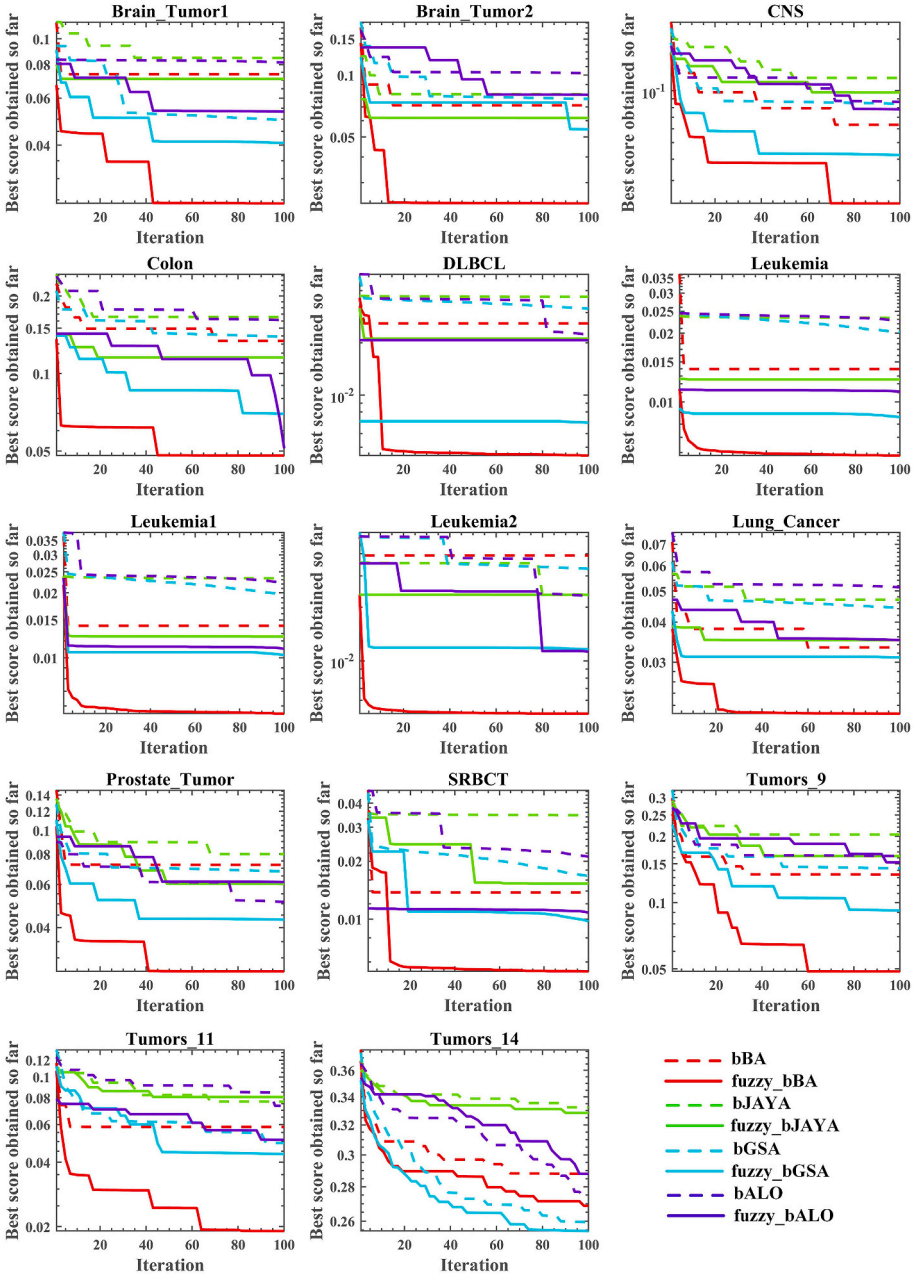


Fig. 5. The convergence curve of the MAs and their fuzzy version.

Table 10
The comparison selected feature number of metaheuristic algorithms with their fuzzy version.

		bBA	fuzzy_bBA	bJAYA	fuzzy_bJAYA	bGSA	fuzzy_bGSA	bALO	fuzzy_bALO
Brain_Tumor1	Avg	1604.8	**492.2**	2792.3	**1379.5**	2386.2	**1326.8**	2657.3	**1446.5**
	Std	**82.416**	383.2834	**66.903**	829.4345	**64.2907**	594.2953	**25.4866**	831.7028
Brain_Tumor2	Avg	3111.2	**982.8**	4911.5	**2142**	4400.5	**2441.9**	4790.5	**2682.4**
	Std	254.6565	**224.579**	**61.1687**	1169.68	**50.8948**	627.8421	**54.9004**	1434.937
CNS	Avg	2278.6	**891.7**	3415.3	**2654**	2975.4	**2154.8**	3274.7	**2746.6**
	Std	**292.958**	408.7344	**53.6968**	1015.358	**66.1366**	544.738	**53.7423**	668.088
Colon	Avg	561.1	**113.4**	933.1	**369**	680.2	**351**	843.3	**331.6**
	Std	**68.7983**	85.8386	**19.3818**	429.3339	**55.9857**	158.3288	**28.1071**	255.0713
DLBCL	Avg	1537.8	**490.3**	2556.4	**1016.4**	2154.5	**758.8**	2448.8	**969.8**
	Std	**134.514**	172.7169	**26.4205**	696.7307	**22.3619**	512.1746	**48.3064**	335.1062
Leukemia	Avg	1988.2	**831.1**	3329.8	**1789.8**	2888.2	**1226.1**	3216.2	**1581.9**
	Std	**73.6294**	271.4285	**15.9708**	767.3922	**44.6089**	738.3407	**25.5421**	596.5092
Leukemia1	Avg	1501.9	**587.7**	2486.8	**1337.9**	2104.7	**1097.8**	2371	**1173.1**
	Std	**66.0058**	103.9092	**20.5199**	574.5476	**17.1078**	246.078	**37.1812**	434.9147
Leukemia2	Avg	3072.3	**1134.8**	5297.8	**2613.9**	4729.2	**2608.1**	5204.2	**2528.3**
	Std	**126.442**	158.0378	**83.6737**	1523.244	**38.0316**	749.2615	**47.6487**	802.4795
Lung_Cancer	Avg	3781.3	**1678.7**	5963.6	**2982.4**	5471.5	**3302.2**	5881.2	**4185**
	Std	376.089	**367.735**	**70.3155**	1798.746	**63.2179**	1046.8649	**67.6179**	1758.143
Prostate_Tumor	Avg	3404.3	**1547.1**	5021.3	**4728.8**	4498.9	**3052.2**	4900.6	**2932.1**
	Std	**472.481**	928.0324	**64.5291**	923.2467	**58.6069**	878.8161	**80.2956**	1654.045
SRBCT	Avg	635.2	**247**	1053	**706.8**	774.8	**449.9**	976.6	**499.1**
	Std	96.2171	**92.4362**	**9.7753**	317.0954	**32.0201**	110.9349	**18.8279**	155.3451
Tumors_9	Avg	2016	**1103.7**	2724.4	**2270**	2341.9	**2002.5**	2626.8	**2089.8**
	Std	**323.598**	430.1994	**38.5175**	777.8539	**45.7443**	380.7023	**26.9353**	709.4693
Tumors_11	Avg	3915.5	**2049.3**	6042.2	**5613.4**	5543.4	**4354.1**	5896	**4489**
	std	471.4385	**375.181**	**96.1755**	1064.756	**118.598**	1104.6329	**84.2101**	1577.893
Tumors_14	avg	5767	**4604.1**	7226.8	**7224**	6971.4	**6441.9**	7241.6	**7004**
	std	**345.313**	897.0077	106.8715	**81.0336**	**146.206**	260.8867	**77.4413**	528.9675
AVG		2512.514	**1196.71**	3839.593	**2630.56**	3422.914	**2254.864**	3737.771	**2475.66**
P-VALUE		**0.00012207**		**0.00012207**		**0.00012207**		**0.00012207**	

Table 11
The comparison time cost of metaheuristic algorithms with their fuzzy version.

		bBA	fuzzy_bBA	bJAYA	fuzzy_bJAYA	bGSA	fuzzy_bGSA	bALO	fuzzy_bALO
Brain_Tumor1	Avg	9.6971E+00	**8.2818E + 00**	**4.6095E + 01**	5.1852E+01	2.1842E+01	**1.9494E + 01**	1.0211E+01	**9.3834E + 00**
	Std	**2.2961E-01**	7.2942E-01	**3.6248E + 00**	8.8739E+00	**2.2171E + 00**	3.1133E+00	**6.4256E-01**	7.4178E-01
Brain_Tumor2	Avg	1.2889E+01	**1.0220E + 01**	6.6742E+01	**6.6655E + 01**	2.5256E+01	**2.4258E + 01**	1.0502E+01	**1.0142E + 01**
	Std	8.9176E-01	**1.2065E-01**	**2.0126E + 00**	5.0950E+00	**7.8274E-01**	2.1533E+00	**2.0079E-01**	1.2777E+00
CNS	Avg	1.0018E+01	**7.6566E + 00**	4.8537E+01	**4.5912E + 01**	1.7617E+01	**1.7439E + 01**	8.3575E+00	**8.1864E + 00**
	Std	8.0229E-01	**1.5781E-01**	2.8088E+00	**1.7605E + 00**	**2.8589E-01**	7.2965E-01	**7.6527E-01**	8.4901E-01
Colon	Avg	2.7447E+00	**2.4874E + 00**	1.4242E+01	**1.3357E + 01**	4.6868E+00	**4.4804E + 00**	2.2663E+00	**2.1231E + 00**
	Std	4.5850E-02	**4.0550E-02**	8.5257E-01	**2.3167E-01**	1.5611E-01	**6.5276E-02**	**1.3230E-02**	5.9939E-02
DLBCL	Avg	7.7263E+00	**7.0088E + 00**	**3.6799E + 01**	3.7603E+01	1.5288E+01	**1.3448E + 01**	8.0043E+00	**6.8445E + 00**
	Std	4.3363E-01	**1.2501E-01**	**3.3572E-01**	5.5816E-01	5.9867E-01	**3.6273E-01**	**4.9357E-02**	2.4383E-01
Leukemia	Avg	1.0049E+01	**8.4731E + 00**	**4.8022E + 01**	4.8518E+01	1.9035E+01	**1.7085E + 01**	9.4104E+00	**8.4444E + 00**
	Std	**1.7644E-01**	2.7516E-01	**6.5969E-01**	8.7631E-01	**4.3270E-01**	5.6909E-01	**1.5730E-01**	4.0404E-01
Leukemia1	Avg	7.3873E+00	**6.6386E + 00**	**3.5454E + 01**	3.6706E+01	1.4481E+01	**1.3153E + 01**	7.4349E+00	**6.4439E + 00**
	Std	**6.9030E-02**	9.7809E-02	1.1302E+00	**3.2394E-01**	**3.8537E-01**	4.2255E-01	**8.2248E-02**	3.7599E-01
Leukemia2	Avg	1.4853E+01	**1.2725E + 01**	**7.0560E + 01**	7.1324E+01	2.8602E+01	**2.6903E + 01**	1.5246E+01	**1.2532E + 01**
	Std	**2.5908E-01**	2.7372E-01	**1.7912E + 00**	2.3320E+00	**5.3419E-01**	6.9497E-01	1.1593E+00	**4.2489E-01**
Lung_Cancer	Avg	3.7940E+01	**3.1157E + 01**	**1.0869E + 02**	1.0919E+02	5.8014E+01	**5.1069E + 01**	4.2158E+01	**3.8755E + 01**
	Std	1.0653E+00	**8.7951E-01**	2.0959E+00	**2.0648E + 00**	**5.4421E-01**	3.2482E+00	**3.4014E-01**	3.1883E+00
Prostate_Tumor	Avg	1.7666E+01	**1.5446E + 01**	7.1939E+01	**7.1597E + 01**	3.0408E+01	**2.8915E + 01**	1.7008E+01	**1.5522E + 01**
	Std	4.8001E-01	**1.2548E + 00**	1.4032E+00	**1.1105E + 00**	**3.6567E-01**	9.0956E-01	**1.0732E-01**	1.1586E+00
SRBCT	Avg	3.8841E+00	**3.5746E + 00**	**1.5541E + 01**	1.6387E+01	6.1418E+00	**5.8197E + 00**	3.4177E+00	**3.2047E + 00**
	Std	9.0344E-02	**6.3415E-02**	**1.7517E-01**	2.0470E-01	**1.0742E-01**	1.1500E-01	**2.6797E-02**	6.7583E-02
Tumors_9	Avg	7.3850E+00	**6.7625E + 00**	**3.7634E + 01**	3.7910E+01	1.4784E+01	**1.4069E + 01**	6.9112E+00	**6.7126E + 00**
	std	4.8294E-01	**2.5960E-01**	6.3981E-01	**5.1810E-01**	**3.3816E-01**	6.8654E-01	**1.5042E-01**	5.5020E-01
Tumors_11	avg	3.2337E+01	**2.7236E + 01**	**1.0083E + 02**	1.0096E+02	5.0303E+01	**4.7929E + 01**	3.4336E+01	**3.3062E + 01**
	std	1.2388E+00	**9.1330E-01**	3.1436E+00	**2.1760E + 00**	**7.2829E-01**	2.9712E+00	**9.0914E-01**	2.6087E+00
Tumors_14	avg	7.9814E+01	**6.8770E + 01**	1.6290E+02	**1.6265E + 02**	**1.0276E + 02**	1.0408E+02	8.3433E+01	**8.1290E + 01**
	std	**3.6479E + 00**	5.2433E+00	**4.5232E + 00**	6.9219E+00	**3.5244E + 00**	6.9977E+00	**2.5989E + 00**	3.0664E+00
AVG		1.8171E+01	**1.5460E + 01**	**6.1713E + 01**	6.2187E+01	2.9230E+01	**2.7724E + 01**	1.8478E+01	**1.7332E + 01**
P-VALUE		**0.0001221**		0.24121		**0.001709**		**0.0001221**	

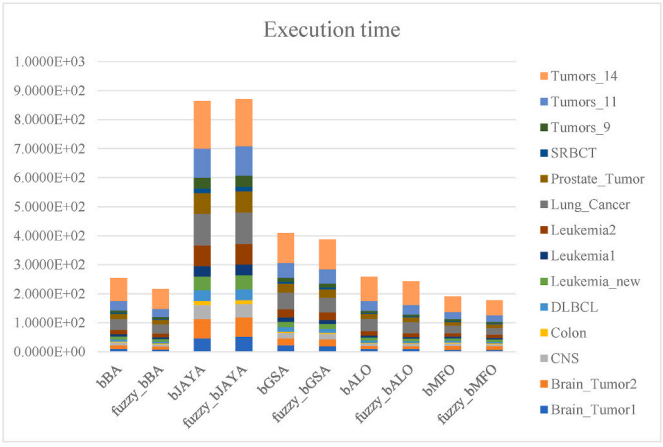


Fig. 6. The comparison of time cost among MAs.

by using our HFRS-initialization on five algorithms. Particularly, in the last experiment, we demonstrated that the fuzzy_bMFO algorithm, which utilizes HFRS-initialization on bMFO, outperforms six currently advanced algorithms. This demonstrates the excellent performance of our algorithm in feature selection problems.

In the future, we plan to continue improving the accuracy of our algorithm. Specifically, we can achieve this by incorporating algorithms with similar complexity but stronger search capabilities for rough set. Additionally, we intend to validate our findings on a broader range of datasets. Furthermore, compared to other filter methods, rough set-based methods inherently have higher time complexity. To address this issue, we can explore combining metaheuristics with other rough set acceleration techniques [103,104] to reduce search time. This could be a promising research direction. Generally, our algorithm performs well in high-dimensional feature selection problems and shows potential in a wide range of areas, including natural language processing, data mining, finance, and transportation. In the end, we can establish an advanced AI framework to handle the majority of feature selection problems.

Declaration of competing interest

This work was supported by the National Key R&D Program of China under Grant Nos. 2021ZD0112501 and 2021ZD0112502; the National Natural Science Foundation of China under Grant Nos. U22A2098, 62172185, 62206105 and 62202200.

Appendix

Table A1
The average fitness results of fuzzy_bMFO using 8 transfer functions

Datasets	metrics	S1	S2	S3	S4	V1	V2	V3	V4
Brain_Tumor1	avg	5.8697E-02	6.3943E-02	7.1989E-02	7.1036E-02	8.7838E-05	5.9966E-05	5.4054E-05	6.7568E-05
	std	5.6492E-02	4.8084E-02	6.7985E-02	6.6430E-02	7.9250E-05	5.9382E-05	5.2095E-05	8.7138E-05
Brain_Tumor2	avg	9.3740E-02	1.0105E-01	9.0066E-02	8.2632E-02	3.1349E-05	1.0128E-04	1.0128E-05	1.0611E-05
	std	1.3184E-01	1.6310E-01	9.9477E-02	9.3615E-02	2.1358E-05	2.1576E-04	3.5587E-06	3.8044E-06
CNS	avg	8.1933E-02	1.1332E-01	9.7462E-02	1.0373E-01	1.3640E-02	8.4163E-05	3.9978E-05	1.6833E-05
	std	8.6762E-02	1.3091E-01	8.4255E-02	1.2113E-01	4.2966E-02	9.0847E-05	3.3563E-05	1.2890E-05
Colon	avg	1.3757E-01	1.4646E-01	1.4312E-01	1.9039E-01	8.2500E-05	6.7500E-05	4.7500E-05	4.7500E-05
	std	1.3362E-01	1.4391E-01	9.0718E-02	1.7081E-01	5.1438E-05	3.3437E-05	1.8447E-05	1.4191E-05
DLBCL	avg	2.5691E-02	2.3970E-02	2.5789E-02	2.3528E-02	3.2913E-05	2.4685E-05	1.8285E-05	1.6456E-05
	std	4.2762E-02	3.7373E-02	3.6872E-02	3.7625E-02	3.4532E-05	2.1136E-05	6.0950E-06	5.7822E-06
Leukemia	avg	1.1866E-02	9.4179E-03	1.0163E-02	1.3076E-02	4.4186E-05	1.9638E-05	1.1923E-05	1.4729E-05
	std	3.2419E-03	6.6514E-03	5.2103E-03	7.8113E-03	7.5975E-05	1.6130E-05	4.7338E-06	5.1751E-06
Leukemia1	avg	1.3379E-02	1.2931E-02	1.5218E-02	1.3844E-02	9.4800E-05	4.6931E-05	3.0974E-05	2.0650E-05
	std	7.2633E-03	4.8640E-03	6.7489E-03	6.3764E-03	7.5925E-05	2.9350E-05	3.2530E-05	5.9363E-06
Leukemia2	avg	2.4331E-02	2.4250E-02	2.4277E-02	2.4313E-02	3.6971E-05	2.4944E-05	1.0690E-05	1.0690E-05
	std	4.2951E-02	3.6988E-02	3.7115E-02	3.9780E-02	2.9326E-05	1.0951E-05	3.1145E-06	3.7562E-06
Lung_Cancer	avg	2.9480E-02	3.8345E-02	3.5525E-02	3.0603E-02	3.1230E-04	4.6560E-03	1.4286E-04	7.0238E-05
	std	3.3280E-02	2.2325E-02	1.9781E-02	3.2724E-02	3.7302E-04	1.4354E-02	1.5028E-04	4.2413E-05
Prostate_Tumor	avg	6.3486E-02	4.8870E-02	5.7278E-02	6.0208E-02	9.0874E-05	9.5852E-03	9.2302E-05	1.9031E-05
	std	4.8394E-02	4.7117E-02	8.5715E-02	4.5060E-02	8.1347E-05	3.0023E-02	2.2887E-04	2.3736E-05
SRBCT	avg	1.4437E-02	2.4544E-02	1.6555E-02	1.7268E-02	8.6655E-05	7.1490E-05	7.1490E-05	5.8492E-05
	std	6.2853E-03	4.1857E-02	6.7602E-03	8.8728E-03	3.2294E-05	1.0465E-05	1.0465E-05	1.7835E-05
Tumors_9	avg	1.9538E-01	1.5353E-01	1.6276E-01	1.5893E-01	7.1024E-02	2.9882E-02	1.2271E-02	2.5742E-02
	std	1.5198E-01	1.3070E-01	9.1964E-02	1.3979E-01	7.7374E-02	6.2152E-02	3.7550E-02	5.3823E-02
Tumors_11	avg	8.1872E-02	8.1285E-02	8.0767E-02	8.1921E-02	6.1794E-03	1.2559E-02	1.1705E-03	6.4224E-03
	std	6.8430E-02	5.1700E-02	5.4286E-02	5.6067E-02	1.6993E-02	2.4165E-02	8.6547E-04	1.7528E-02
Tumors_14	avg	2.9886E-01	3.0895E-01	3.1976E-01	3.1628E-01	1.8266E-01	2.0008E-01	1.8540E-01	1.6581E-01
	std	6.9590E-02	8.6108E-02	6.5782E-02	6.5448E-02	4.5936E-02	7.5028E-02	6.0500E-02	6.3190E-02
RANK-ARV		6.384615	6.307692	6.538462	6.769231	3.384615	3.307692	1.692308	1.384615
RANK		6	5	7	8	4	3	2	1

Table A2
The average error rate results of fuzzy_bMFO using 8 transfer functions

Datasets	metrics	S1	S2	S3	S4	V1	V2	V3	V4
Brain_Tumor1	avg	4.4444E-02	5.3333E-02	6.4444E-02	6.4444E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	std	5.7378E-02	5.6364E-02	7.2917E-02	7.2917E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Brain_Tumor2	avg	8.1667E-02	9.5000E-02	7.9286E-02	7.5952E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

(*continued on next page*)

Table A2 (*continued*)

Datasets	metrics	S1	S2	S3	S4	V1	V2	V3	V4
	std	1.4282E-01	1.7070E-01	1.0543E-01	1.0165E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
CNS	avg	6.6667E-02	1.0000E-01	8.4286E-02	8.7619E-02	1.4286E-02	0.0000E+00	0.0000E+00	0.0000E+00
	std	8.6066E-02	1.4055E-01	8.9877E-02	1.3300E-01	4.5175E-02	0.0000E+00	0.0000E+00	0.0000E+00
Colon	avg	1.2381E-01	1.3095E-01	1.2619E-01	1.7619E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	std	1.4010E-01	1.5276E-01	9.5271E-02	1.8000E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
DLBCL	avg	1.4286E-02	1.2500E-02	1.2500E-02	1.2500E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	std	4.5175E-02	3.9528E-02	3.9528E-02	3.9528E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Leukemia	avg	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	std	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Leukemia1	avg	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	std	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Leukemia2	avg	1.4286E-02	1.2500E-02	1.2500E-02	1.4286E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	std	4.5175E-02	3.9528E-02	3.9528E-02	4.5175E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Lung_Cancer	avg	1.5288E-02	2.4762E-02	1.9286E-02	1.4286E-02	0.0000E+00	4.7619E-03	0.0000E+00	0.0000E+00
	std	3.4977E-02	2.6111E-02	2.4907E-02	3.2140E-02	0.0000E+00	1.5058E-02	0.0000E+00	0.0000E+00
Prostate_Tumor	avg	4.9091E-02	2.9091E-02	3.7273E-02	4.9091E-02	0.0000E+00	1.0000E-02	0.0000E+00	0.0000E+00
	std	5.1817E-02	4.6906E-02	8.8498E-02	5.1817E-02	0.0000E+00	3.1623E-02	0.0000E+00	0.0000E+00
SRBCT	avg	0.0000E+00	1.2500E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	std	0.0000E+00	3.9528E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Tumors_9	avg	1.8540E-01	1.3857E-01	1.4714E-01	1.4595E-01	7.4167E-02	3.0952E-02	1.2500E-02	2.6786E-02
	std	1.6634E-01	1.3862E-01	9.6616E-02	1.4778E-01	8.1408E-02	6.5494E-02	3.9528E-02	5.6626E-02
Tumors_11	avg	6.3245E-02	6.6105E-02	6.2499E-02	6.3455E-02	5.5556E-03	1.2132E-02	0.0000E+00	5.8824E-03
	std	7.0654E-02	5.4718E-02	5.4375E-02	5.4855E-02	1.7568E-02	2.5592E-02	0.0000E+00	1.8602E-02
Tumors_14	avg	2.9055E-01	2.9908E-01	3.1055E-01	3.0684E-01	1.8822E-01	2.0576E-01	1.9253E-01	1.7221E-01
	std	7.3777E-02	9.0882E-02	6.9260E-02	6.8989E-02	4.8458E-02	7.9756E-02	6.3608E-02	6.6191E-02
Rank-ARV		5.142857	5.785714	5.071429	5.285714	1.571429	2	1.142857	1.214286
Rank		6	8	5	7	3	4	1	2

Table A3
The average selected feature number of fuzzy_bMFO using 8 transfer functions

Datasets	metrics	S1	S2	S3	S4	V1	V2	V3	V4
Brain_Tumor1	avg	1950.6	1571.9	1274.8	1162	10.4	7.1	6.4	8
	std	1028.288	985.2794	960.1515	1048.276	9.3832	7.0309	6.168	10.3172
Brain_Tumor2	avg	3350	2238.8	3057.2	2172.3	6.5	21	2.1	2.2
	std	1374.978	585.8923	1814.736	1202.742	4.4284	44.7363	0.73786	0.78881
CNS	avg	2651.9	2612.2	2479.6	2921.7	9.8	12	5.7	2.4
	std	1231.22	1020.774	1290.61	1196.941	9.2352	12.9529	4.7854	1.8379
Colon	avg	798.1	882.2	929.7	920.4	3.3	2.7	1.9	1.9
	std	156.4569	97.5019	60.2348	22.3468	2.0575	1.3375	0.73786	0.56765
DLBCL	avg	1325.6	1322.9	1521.9	1274.6	3.6	2.7	2	1.8
	std	716.3159	739.0543	714.7818	729.4603	3.7771	2.3118	0.66667	0.63246
Leukemia	avg	1691.8	1342.8	1449	1864.4	6.3	2.8	1.7	2.1
	std	462.225	948.3589	742.8801	1113.736	10.8326	2.2998	0.67495	0.73786
Leukemia1	avg	1425.4	1377.7	1621.3	1474.9	10.1	5	3.3	2.2
	std	773.8355	518.208	719.0301	679.3383	8.0891	3.1269	3.4657	0.63246
Leukemia2	avg	2415.6	2778.1	2784.2	2411.5	8.3	5.6	2.4	2.4
	std	1290.211	1179.157	1198.643	1773.856	6.5836	2.4585	0.69921	0.84327
Lung_Cancer	avg	3768.9	3735	4335.3	4291.9	78.7	33.3	36	17.7
	std	1623.276	2247.863	2228.252	1886.683	94.0001	20.9923	37.8711	10.688
Prostate_Tumor	avg	3541.4	4462.8	4596.4	2852.5	19.1	17.9	19.4	4
	std	1706.33	1240.989	1611.653	2021.368	17.0974	14.4103	48.1045	4.9889
SRBCT	avg	666.4	584.8	764.2	797.1	4	3.3	3.3	2.7
	std	290.1315	274.4719	312.0494	409.569	1.4907	0.48305	0.48305	0.82327
Tumors_9	avg	2205.3	2506.6	2631.1	2321.4	64.8	54.7	45.4	33.9
	std	926.9359	577.2362	503.9972	800.1636	37.7736	56.0418	43.226	47.1085
Tumors_11	avg	5461.7	4633.7	5362.4	5424	226	259.1	293.4	209.1
	std	934.8181	1561.338	1514.525	1532.73	147.2051	306.748	216.9399	139.9511
Tumors_14	avg	6855.5	7451.3	7425.9	7440	1154.7	1382.4	750.5	664.3
	std	907.0797	122.2166	71.1609	66.3124	305.9245	650.1448	235.0306	219.4883
Rank-ARV		6.461538	6.153846	7	6.384615	3.538462	3	1.923077	1.307692
Rank		7	5	8	6	4	3	2	1

Table A4
The average computation time of fuzzy_bMFO using 8 transfer functions

Datasets	metrics	S1	S2	S3	S4	V1	V2	V3	V4
Brain_Tumor1	avg	1.07911E+01	1.04498E+01	1.05459E+01	1.07556E+01	9.12210E+00	6.77800E+00	6.73980E+00	6.59670E+00
	std	4.67110E-01	5.42700E-02	4.74240E-02	1.15040E-01	1.79610E-01	2.33950E-01	1.41430E-01	1.04670E-01
Brain_Tumor2	avg	1.18521E+01	1.48993E+01	1.53715E+01	1.40231E+01	1.45405E+01	1.02720E+01	1.02898E+01	1.00631E+01
	std	1.54310E-01	2.42370E-01	3.16100E-01	9.19000E-01	8.89420E-01	3.57740E-01	1.88330E-01	1.54760E-01
CNS	avg	1.15569E+01	1.13580E+01	1.13516E+01	1.12495E+01	1.01640E+01	7.31560E+00	7.05680E+00	7.16410E+00
	std	1.52330E-01	1.44490E-01	9.53500E-02	9.59160E-01	3.51390E-01	3.54610E-01	2.01900E-01	1.57500E-01
Colon	avg	2.86880E+00	2.79130E+00	2.90890E+00	2.86040E+00	3.18090E+00	2.29280E+00	2.33420E+00	2.32750E+00
	std	1.16190E-01	4.47100E-02	8.17440E-02	1.06450E-01	1.09740E-01	5.84440E-02	9.02930E-02	1.02160E-01
DLBCL	avg	9.73540E+00	1.01975E+01	9.60190E+00	1.00944E+01	9.03870E+00	6.02160E+00	6.22890E+00	6.04400E+00
	std	3.17590E-01	5.38970E-01	5.40260E-01	4.14730E-01	3.97430E-01	3.04310E-01	4.87250E-01	3.57700E-01
Leukemia	avg	1.07306E+01	1.07066E+01	1.08589E+01	1.08790E+01	1.00614E+01	7.70020E+00	7.28750E+00	7.02820E+00
	std	1.37050E-01	1.57810E-01	1.42150E-01	2.08510E-01	2.41380E-01	8.95330E-01	1.04050E-01	9.68030E-02
Leukemia1	avg	8.40460E+00	8.53870E+00	8.55300E+00	8.97140E+00	7.93860E+00	5.77000E+00	5.61930E+00	5.57490E+00
	std	1.14570E-01	1.49820E-01	1.19110E-01	5.68050E-01	1.68030E-01	1.68270E-01	1.09080E-01	6.68510E-02
Leukemia2	avg	1.61205E+01	1.63559E+01	1.67794E+01	1.65922E+01	1.51474E+01	1.05138E+01	1.03679E+01	1.04672E+01
	std	2.14700E-01	2.58980E-01	1.15660E+00	6.80960E-01	2.04640E-01	1.81620E-01	1.62720E-01	1.11450E-01
Lung_Cancer	avg	4.40039E+01	4.41460E+01	4.50374E+01	4.41107E+01	3.28061E+01	2.35346E+01	2.27535E+01	2.18671E+01
	std	9.35250E-01	3.92400E-01	4.17370E-01	5.03890E-01	2.03790E+00	1.02700E+00	1.24080E+00	5.94830E-01
Prostate_Tumor	avg	1.86929E+01	1.88852E+01	1.90173E+01	1.93186E+01	1.63836E+01	1.22163E+01	1.19886E+01	1.15908E+01
	std	5.96820E-02	7.23290E-02	6.04810E-02	6.40920E-01	3.47770E-01	4.12110E-01	4.24680E-01	2.49680E-01
SRBCT	avg	3.88200E+00	3.88970E+00	3.95570E+00	3.90780E+00	4.06610E+00	3.02070E+00	3.03290E+00	3.00410E+00
	std	3.52050E-02	2.08320E-02	4.02290E-02	1.99650E-02	1.65850E-01	3.52580E-02	2.95220E-02	3.97450E-02
Tumors_9	avg	7.69860E+00	7.85180E+00	7.85900E+00	7.79910E+00	8.13610E+00	5.75870E+00	5.64810E+00	5.54110E+00
	std	1.30040E-01	2.39520E-01	1.47700E-01	1.34510E-01	2.34690E-01	1.89400E-01	1.90020E-01	1.59600E-01
Tumors_11	avg	3.58040E+01	3.83411E+01	4.04980E+01	3.93933E+01	2.90006E+01	2.36457E+01	2.33363E+01	2.10680E+01
	std	8.96040E-01	2.96530E+00	3.34740E+00	4.02790E+00	2.71280E+00	2.38760E+00	2.29300E+00	1.06050E+00
Tumors_14	avg	9.52926E+01	9.02555E+01	8.66715E+01	8.57242E+01	6.27364E+01	5.51144E+01	4.94521E+01	4.43985E+01
	std	4.97160E+00	4.50820E+00	2.53870E+00	3.03110E+00	4.97100E+00	5.24900E+00	3.29490E+00	2.88550E+00
Rank-ARV		5.642857	6.071429	6.928571	6.357143	5	2.571429	2.142857	1.285714
Rank		5	6	8	7	4	3	2	1

Table A5
comparison error of bMFO initialized with RS-based methods and other filter-based methods

		bMFO	fuzzy_bMFO	bWOA_bMFO	bDE_bMFO	bPSO_bMFO	MRMR_bMFO	reliefF_bMFO
Brain_Tumor1	Avg	1.0000E-02	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	1.0000E-02	**0.0000E + 00**
	Std	3.1623E-02	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	3.1623E-02	**0.0000E + 00**
Brain_Tumor2	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
CNS	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Colon	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
DLBCL	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Leukemia	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Leukemia1	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	Std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Leukemia2	Avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Lung_Cancer	avg	5.0000E-03	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	4.7619E-03	**0.0000E + 00**	**0.0000E + 00**
	std	1.5811E-02	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	1.5058E-02	**0.0000E + 00**	**0.0000E + 00**
Prostate_Tumor	avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
SRBCT	avg	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
	std	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**	**0.0000E + 00**
Tumors_9	avg	3.0952E-02	**0.0000E + 00**	3.4286E-02	**0.0000E + 00**	2.8571E-02	2.0000E-02	1.4286E-02
	std	6.5494E-02	**0.0000E + 00**	7.3525E-02	**0.0000E + 00**	6.0234E-02	6.3246E-02	4.5175E-02
Tumors_11	avg	4.7619E-03	**0.0000E + 00**	5.8824E-03	1.1438E-02	1.0882E-02	5.8824E-03	**0.0000E + 00**
	std	1.5058E-02	**0.0000E + 00**	1.8602E-02	2.4126E-02	2.3036E-02	1.8602E-02	**0.0000E + 00**
Tumors_14	avg	1.9469E-01	**1.7872E-01**	1.9182E-01	1.9782E-01	1.8804E-01	1.8121E-01	1.9690E-01
	std	6.2629E-02	**7.7273E-02**	7.5472E-02	5.4566E-02	7.0849E-02	4.9281E-02	6.9053E-02
AVG		1.7529E-02	**1.2766E-02**	1.6571E-02	1.4947E-02	1.6590E-02	1.5507E-02	1.5085E-02
RANK-ARV		2.57142857	**1**	1.85714286	1.85714286	2.14285714	1.857142857	1.5
RANK		7	**1**	3	3	6	3	2

Table A6
comparison selected feature number of bMFO initialized with RS-based methods and other filter-based methods

		bMFO	fuzzy_bMFO	bWOA_bMFO	bDE_bMFO	bPSO_bMFO	MRMR_bMFO	reliefF_bMFO
Brain_Tumor1	Avg	13.1	8.7	5.7	9.9	5.3	**3.7**	5.4
	Std	22.8641	10.1549	5.4782	16.1758	3.164	**1.7029**	4.4771
Brain_Tumor2	Avg	7.9	2.8	**2.3**	4.7	3.1	2.6	2.6
	Std	11.1699	1.6193	**0.67495**	7.5432	2.4698	1.2649	1.1738
CNS	Avg	4.4	2.6	3.3	2.1	4.3	**1.8**	3.7
	Std	4.6714	2.1705	3.3015	0.99443	2.3594	**0.91894**	1.767
Colon	Avg	2	1.6	**1.5**	1.6	1.7	1.7	2
	Std	0.94281	0.69921	0.52705	0.5164	0.67495	**0.48305**	0.66667
DLBCL	Avg	1.8	**1.4**	1.7	1.5	2	**1.4**	1.6
	std	0.63246	0.5164	0.94868	0.70711	**0.4714**	0.5164	0.5164
Leukemia	avg	1.9	1.6	1.4	1.4	1.7	**1**	1.4
	std	0.56765	0.5164	0.5164	0.5164	0.82327	**0**	0.5164
Leukemia1	avg	2.6	2.2	**2.1**	2.8	4	2.2	2.2
	std	1.5776	**0.78881**	0.8756	2.3476	6	1.1353	0.78881
Leukemia2	avg	2.7	2.4	2.4	2.9	2.9	**1.8**	1.9
	std	1.1595	0.96609	0.84327	0.8756	1.2867	0.63246	**0.31623**
Lung_Cancer	avg	43.6	21.9	29.5	**13.2**	30.5	33.2	32.3
	std	42.7348	18.248	26.4921	**8.8919**	21.7932	44.5291	39.2628
Prostate_Tumor	avg	7.8	4.8	7.8	7.8	8.8	**2.7**	6.9
	std	7.4506	3.5839	11.6886	11.2724	9.8635	**1.3375**	9.4687
SRBCT	avg	2.9	2.7	3.5	3.7	3.2	**2.6**	2.9
	std	**0.56765**	0.82327	1.2693	1.0593	0.91894	0.69921	1.3703
Tumors_9	avg	65	83.1	**33.4**	165.5	75.4	68.8	61
	std	52.0683	76.4903	**40.6535**	131.6884	66.8783	73.4193	41.841
Tumors_11	avg	340	373.2	**263.8**	342.7	301.6	336.2	398.2
	std	172.2085	226.1233	**115.847**	242.7573	185.4222	211.2885	264.5759
Tumors_14	avg	2097.1	1621.3	1836.8	2006.9	2190.4	**1518.1**	1721.2
	std	908.9571	844.9396	1056.983	1151.891	1115.743	1037.039	**783.194**
AVG		185.2	152.1643	156.8	183.3357	188.2071	**141.271**	160.2357
RANK-ARV		5.5	3.214286	2.857143	4.428571	5.214286	**2.07143**	3.5
RANK		7	3	2	5	6	**1**	4

Table A7
comparison time of bMFO initialized with RS-based methods and other filter-based methods

		bMFO	fuzzy_bMFO	bWOA_bMFO	bDE_bMFO	bPSO_bMFO	MRMR_bMFO	reliefF_bMFO
Brain_Tumor1	Avg	7.6369E+00	7.6087E+00	6.4272E+00	6.4835E+00	6.5511E+00	**6.3732E + 00**	6.4484E+00
	Std	7.0883E-01	1.2459E+00	1.3960E-01	1.5516E-01	**8.4153E-02**	1.3854E-01	1.1163E-01
Brain_Tumor2	Avg	1.3273E+01	1.2329E+01	9.7171E+00	9.7146E+00	9.4625E+00	1.0352E+01	**9.0455E + 00**
	Std	1.8161E+00	2.0881E+00	**6.2668E-01**	1.1840E+00	1.4340E+00	1.0500E+00	1.0852E+00
CNS	Avg	7.8539E+00	6.5909E+00	6.3299E+00	6.9857E+00	7.3242E+00	6.3999E+00	**6.2788E + 00**
	Std	1.6685E+00	2.9042E-01	8.7098E-02	1.3775E+00	1.3074E+00	1.4813E-01	**6.4530E-02**
Colon	Avg	3.4654E+00	2.3174E+00	2.0453E+00	2.0591E+00	2.1663E+00	2.0482E+00	**2.0418E + 00**
	Std	1.5889E+00	2.3124E-01	5.0455E-02	3.3534E-02	4.1439E-02	5.5030E-02	**3.0266E-02**
DLBCL	Avg	6.1378E+00	5.8781E+00	6.0335E+00	**5.5068E + 00**	5.6855E+00	5.6189E+00	5.5421E+00
	Std	2.4962E-01	2.5312E-01	5.2295E-01	**5.6798E-02**	7.4349E-02	1.4957E-01	6.4719E-02
Leukemia	Avg	8.5587E+00	7.6801E+00	7.0561E+00	7.2246E+00	6.9004E+00	**6.6360E + 00**	8.0002E+00
	std	1.4091E+00	6.4290E-01	7.9164E-02	1.0374E-01	6.3291E-02	**4.4545E-02**	1.7043E+00
Leukemia1	avg	6.2098E+00	5.6155E+00	5.3989E+00	7.2351E+00	5.7143E+00	**5.3429E + 00**	5.9438E+00
	std	9.8002E-01	1.5306E-01	**7.4009E-02**	1.0303E+00	3.3244E-01	1.3009E-01	5.6219E-01
Leukemia2	avg	1.0996E+01	1.1445E+01	1.0399E+01	1.1176E+01	1.1056E+01	1.0459E+01	**1.0232E + 01**
	std	4.2554E-01	6.8946E-01	**1.8362E-01**	7.1606E-01	2.7441E-01	3.1819E-01	2.8888E-01
Lung_Cancer	avg	2.5833E+01	2.2169E+01	2.3367E+01	2.3585E+01	2.5769E+01	**2.2013E + 01**	2.2533E+01
	std	3.7404E+00	**9.5745E-01**	1.2145E+00	1.1906E+00	1.7891E+00	1.2384E+00	1.2174E+00
Prostate_Tumor	avg	1.2463E+01	1.2346E+01	1.1890E+01	**1.1884E + 01**	1.2925E+01	1.2781E+01	1.2093E+01
	std	4.3900E-01	4.7535E-01	4.3699E-01	**3.0953E-01**	8.4030E-01	6.2226E-01	5.4649E-01
SRBCT	avg	3.3113E+00	3.2430E+00	3.2043E+00	3.1503E+00	**3.1116E + 00**	3.2778E+00	3.1738E+00
	std	1.1416E-01	**5.4758E-02**	1.3839E-01	8.0460E-02	8.4801E-02	1.0795E-01	1.5339E-01
Tumors_9	avg	6.9454E+00	6.2052E+00	6.3224E+00	**5.7113E + 00**	6.3504E+00	5.8060E+00	6.1198E+00
	std	7.0491E-01	**1.3238E-01**	2.7700E-01	2.8785E-01	5.9545E-01	3.9042E-01	1.2334E+00
Tumors_11	avg	2.4119E+01	2.2549E+01	2.2759E+01	2.2706E+01	2.3918E+01	**2.1258E + 01**	2.3887E+01
	std	1.9696E+00	1.1377E+00	2.1495E+00	1.4421E+00	2.0635E+00	**7.6347E-01**	1.9100E+00
Tumors_14	avg	5.4686E+01	5.1435E+01	4.7413E+01	5.2529E+01	5.0142E+01	**4.4062E + 01**	4.4912E+01
	std	5.4963E+00	3.1833E+00	3.4163E+00	3.9534E+00	4.5261E+00	4.9128E+00	**2.9884E + 00**
AVG		1.3678E+01	1.2672E+01	1.2026E+01	1.2568E+01	1.2648E+01	**1.1602E + 01**	1.1875E+01
RANK-ARV		6.5714	4.5714	3.2143	3.7143	4.5000	**2.6429**	2.7857
RANK		7	6	3	4	5	**1**	2

Table A8
comparation RS dependency of MAs and other filters

	random	bWOA-variant	bWOA	bDE	bPSO	MRMR	reliefF
Brain_Tumor1	**1**	**1**	**1**	**1**	**1**	0.92222	0.033333
Brain_Tumor2	**1**	**1**	**1**	**1**	**1**	0.64	0.04
CNS	**1**	**1**	**1**	**1**	**1**	0.78333	0
Colon	**1**	**1**	**1**	**1**	**1**	0.93548	0
DLBCL	**1**	**1**	**1**	**1**	**1**	0.98701	0.50649
Leukemia	**1**	**1**	**1**	**1**	**1**	0.93056	0.88889
Leukemia1	**1**	**1**	**1**	**1**	**1**	0.79167	0.375
Leukemia2	**1**	**1**	**1**	**1**	**1**	0.79167	0
Lung_Cancer	**1**	**1**	**1**	**1**	**1**	0.95567	0.26108
Prostate_Tumor	**1**	**1**	**1**	**1**	**1**	0.92157	0.058824
SRBCT	**1**	**1**	**1**	**1**	**1**	0.92771	0.31325
Tumors_9	**1**	**1**	**1**	**1**	**1**	0.85	0.016667
Tumors_11	**1**	**1**	**1**	**1**	**1**	0.98851	0.017241
Tumors_14	**1**	**1**	0.99675	**1**	**1**	0.96104	0
AVG	**1**	**1**	0.999768	**1**	**1**	0.884746	0.179341
RANK-ARV	**1**	**1**	1.285714	**1**	**1**	6	7
RANK	**1**	**1**	5	**1**	**1**	6	7

Table A9
comparation RS feature selected rate of MAs and other filters

	random	bWOA-variant	bWOA	bDE	bPSO	MRMR	reliefF
Brain_Tumor1	0.49899	0.002872	0.004899	0.099831	0.48851	0.002872	**0.00017**
Colon	0.517	0.003	0.004	0.0755	0.481	0.003	**0.0005**
Brain_Tumor2	0.49667	0.001447	0.002219	0.009743	0.49465	0.001447	**9.65E-05**
CNS	0.49909	0.002244	0.002385	0.068593	0.4932	0.002244	**0.00014**
DLBCL	0.50539	0.003291	0.003108	0.010422	0.48802	0.003291	**0.00018**
Leukemia	0.49586	0.001964	0.001824	0.032964	0.49179	0.001964	**0.00042**
Leukemia1	0.50047	0.002253	0.002253	0.012577	0.48188	0.002253	**0.00019**
Leukemia2	0.50993	0.001425	0.001247	0.066548	0.48713	0.001425	**8.91E-05**
Lung_Cancer	0.50175	0.001905	0.003016	0.045238	0.49357	0.001905	**7.94E-05**
Prostate_Tumor	0.499	0.003616	0.005709	0.027405	0.48939	0.003616	**9.52E-05**
SRBCT	0.513	0.005633	0.005633	0.052426	0.47097	0.005633	**0.00043**
Tumors_9	0.50786	0.002445	0.00227	0.074048	0.49319	0.002445	**0.00017**
Tumors_11	0.49964	0.002474	0.002872	0.063672	0.49374	0.002474	**7.98E-05**
Tumors_14	0.49777	0.048438	0.089013	0.093944	0.4921	0.048438	**6.66E-05**
AVG	0.50303	0.005929	0.009318	0.052351	0.48851	0.005929	**0.00019**
RANK-ARV	7	2.285714	3.142857	5	6	2.285714	**1**
RANK	7	2	4	5	6	2	**1**

References

[1] Amparo Alonso-Betanzos, et al., A review of microarray datasets: where to find them and specific characteristics, Microarray Bioinformatics (2019) 65–85.
[2] Gregory Piatetsky-Shapiro, Pablo Tamayo, Microarray data mining: facing the challenges, ACM SIGKDD Explorations Newsletter 5 (2) (2003) 1–5.
[3] Todd R. Golub, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Sci. Technol. Humanit. 286 (5439) (1999) 531–537.
[4] Manoranjan Dash, Hua Liu, Jun Yao, Dimensionality reduction of unsupervised data, in: Proceedings Ninth Ieee International Conference on Tools with Artificial Intelligence, IEEE, 1997.
[5] Anil Jain, Zongker Douglas, Feature selection: evaluation, application, and small sample performance, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 153–158.
[6] Isabelle Guyon, et al. (Eds.), Feature Extraction: Foundations and Applications vol. 207, Springer, 2008.
[7] Hanchuan Peng, Fuhui Long, Chris Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[8] Igor Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: European Conference on Machine Learning, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.
[9] Mark A. Hall, Correlation-based Feature Selection for Machine Learning, Diss. The University of Waikato, 1999.
[10] Lei Yu, Huan Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003.
[11] Isabelle Guyon, et al., Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[12] Sebastián Maldonado, Richard Weber, Jayanta Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Inf. Sci. 181 (1) (2011) 115–128.
[13] Zhiqing Chen, et al., Dispersed differential hunger games search for high dimensional gene data feature selection, Comput. Biol. Med. (2023), 107197.
[14] Zhiqing Chen, et al., An artificial bee bare-bone hunger games search for global optimization and high-dimensional feature selection, iScience 26 (2023) 5.
[15] Gehad Ismail Sayed, Mona M. Soliman, Aboul Ella Hassanien, A novel melanoma prediction model for imbalanced data using optimized SqueezeNet by bald eagle search optimization, Comput. Biol. Med. 136 (2021), 104712.
[16] Jeremy Kawahara, Aicha BenTaieb, Ghassan Hamarneh, Deep features to classify skin lesions, in: 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), IEEE, 2016.
[17] Asunción Jiménez-Cordero, Juan Miguel Morales, Salvador Pineda, A novel embedded min-max approach for feature selection in nonlinear support vector machine classification, Eur. J. Oper. Res. 293 (1) (2021) 24–35.
[18] Talbi El-Ghazali, Metaheuristics: from Design to Implementation, John Wiley & Sons, 2009.
[19] Soumitri Chattopadhyay, Aritra Marik, Rishav Pramanik, A Brief Overview of Physics-Inspired Metaheuristic Optimization Techniques, 2022 arXiv preprint arXiv:2201.12810.
[20] Iztok Fister Jr., et al., A Brief Review of Nature-Inspired Algorithms for Optimization, 2013 *arXiv preprint arXiv:1307.4186*.
[21] Xin-She Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 65–74.
[22] R. Rao, Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems, Int. J. Ind. Eng. Comput. 7 (1) (2016) 19–34.
[23] Esmat Rashedi, Hossein Nezamabadi-Pour, Saeid Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.

[24] Seyedali Mirjalili, The ant lion optimizer, Adv. Eng. Software 83 (2015) 80–98.
[25] Seyedali Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowl. Base Syst. 89 (2015) 228–249.
[26] Shimin Li, et al., Slime mould algorithm: a new method for stochastic optimization, Future Generat. Comput. Syst. 111 (2020) 300–323.
[27] Yutao Yang, et al., Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, Expert Syst. Appl. 177 (2021), 114864.
[28] Iman Ahmadianfar, et al., RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method, Expert Syst. Appl. 181 (2021), 115079.
[29] Jiaze Tu, et al., The colony predation algorithm, JBE 18 (2021) 674–710.
[30] Iman Ahmadianfar, et al., INFO: an efficient optimization algorithm based on weighted mean of vectors, Expert Syst. Appl. 195 (2022), 116516.
[31] Ali Asghar Heidari, et al., Harris hawks optimization: algorithm and applications, Future Generat. Comput. Syst. 97 (2019) 849–872.
[32] Hang Su, et al., RIME: a physics-based optimization, Neurocomputing 532 (2023) 183–214.
[33] Manosij Ghosh, et al., A wrapper-filter feature selection technique based on ant colony optimization, Neural Comput. Appl. 32 (2020) 7839–7857.
[34] Pablo Bermejo, Jose A. Gámez, Jose M. Puerta, A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets, Pattern Recogn. Lett. 32 (5) (2011) 701–711.
[35] Ritam Guha, et al., Embedded chaotic whale survival algorithm for filter–wrapper feature selection, Soft Comput. 24 (17) (2020) 12821–12843.
[36] Adrian Pino Angulo, Kilho Shin, Mrmr+ and Cfs+ feature selection algorithms for high-dimensional data, Appl. Intell. 49 (2019) 1954–1967.
[37] Shankar Thawkar, et al., Breast cancer prediction using a hybrid method based on butterfly optimization algorithm and ant lion optimizer, Comput. Biol. Med. 139 (2021), 104968.
[38] Ali El Akadi, et al., A new gene selection approach based on Minimum Redundancy-Maximum Relevance (MRMR) and Genetic Algorithm (GA), in: 2009 IEEE/ACS International Conference on Computer Systems and Applications, IEEE, 2009.
[39] Sandhya Rani Bansal, Savita Wadhawan, Rajeev Goel, mrmr-pso: a hybrid feature selection technique with a multiobjective approach for sign language recognition, Arabian J. Sci. Eng. 47 (8) (2022) 10365–10380.
[40] Shreem, Salam Salameh, et al., Hybridizing ReliefF, MRMR filters and GA wrapper approaches for gene selection, J. Theor. Appl. Inf. Technol. 46 (2) (2012) 1034–1039.
[41] H. Salem, G. Attiya, N. El-Fishawy, Classification of human cancer diseases by gene expression profiles, Appl. Soft Comput. 50 (2017) 124–134.
[42] Santos Kumar Baliarsingh, Swati Vipsita, Chaotic emperor penguin optimised extreme learning machine for microarray cancer classification, IET Syst. Biol. 14 (2) (2020) 85–95.
[43] Santos Kumar Baliarsingh, Swati Vipsita, Bodhisattva Dash, A new optimal gene selection approach for cancer classification using enhanced Jaya-based forest optimization algorithm, Neural Comput. Appl. 32 (2020) 8599–8616.
[44] A.K. Shukla, Feature Selection Inspired by Human Intelligence for Improving Classification Accuracy of Cancer Types, Computational Intelligence, 2020.
[45] Borhan Kazimipour, Xiaodong Li, A. Kai Qin, A review of population initialization techniques for evolutionary algorithms, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014.
[46] Shahryar Rahnamayan, Hamid R. Tizhoosh, Magdy MA. Salama, A novel population initialization method for accelerating evolutionary algorithms, Comput. Math. Appl. 53 (10) (2007) 1605–1614.
[47] Hajira Jabeen, Zunera Jalil, Abdul Rauf Baig, Opposition based initialization in particle swarm optimization (O-PSO), in: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, 2009.
[48] Millie Pant, Radha Thangaraj, Ajith Abraham, Particle swarm optimization: performance tuning and empirical analysis, in: Foundations of Computational Intelligence vol. 3, Global Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 101–128.
[49] Nguyen Quang Uy, et al., Initialising PSO with randomised low-discrepancy sequences: the comparative results, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007.
[50] Borhan Kazimipour, Xiaodong Li, A. Kai Qin, Initialization methods for large scale global optimization, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013.
[51] Yu Xue, Weiwei Jia, Alex X. Liu, A particle swarm optimization with filter-based population initialization for feature selection, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019.
[52] Yu Xue, Xu Cai, Weiwei Jia, Particle swarm optimization based on filter-based population initialization method for feature selection in classification, J. Ambient Intell. Hum. Comput. 14 (6) (2023) 7355–7366.
[53] Li-Xin Zhang, et al., A novel hybrid feature selection algorithm: using ReliefF estimation for GA-Wrapper search, in: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693) vol. 1, IEEE, 2003.
[54] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer Academic Publisher, London, 1991.
[55] Aboul Ella Hassanien, et al., An improved moth flame optimization algorithm based on rough sets for tomato diseases detection, Comput. Electron. Agric. 136 (2017) 86–96.
[56] Pradipta Maji, et al., Rough sets for Insilico identification of differentially expressed miRNAs, in: Scalable Pattern Recognition Algorithms: Applications in Computational Biology and Bioinformatics, 2014, pp. 171–193.
[57] Zilin Zeng, et al., A hybrid feature selection method based on rough conditional mutual information and naive Bayesian Classifier, Int. Sch. Res. Notices (2014), 2014.
[58] Fatemeh Aghaeipoor, Mohammad Masoud Javidi, A hybrid fuzzy feature selection algorithm for high-dimensional regression problems: an mRMR-based framework, Expert Syst. Appl. 162 (2020), 113859.
[59] Shuyin Xia, et al., An efficient and accurate rough set for feature selection, classification, and knowledge representation, IEEE Trans. Knowl. Data Eng. (2022) 7724 - 7735.
[60] Didier Dubois, Henri Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. Gen. Syst. 17 (1990) 2–3, 191-209.
[61] Didier Dubois, Henri Prade, Putting rough sets and fuzzy sets together, in: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Springer Netherlands, Dordrecht, 1992, pp. 203–232.
[62] Qinghua Hu, Zongxia Xie, Daren Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recogn. 40 (12) (2007) 3509–3521.
[63] A. Skowron, C. Rauszer, 1The discernibility matrices and functions in information systems, in: R. Słowiński (Ed.), Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, Springer Netherlands, Dordrecht, 1992, pp. 331–362.
[64] Haleh Vafaie, Ibrahim F. Imam, Feature selection methods: genetic algorithms vs. greedy-like search, Proceedings of the international conference on fuzzy and intelligent control systems 51 (1994).
[65] Xiaohua Hu, Nick Cercone, Jiawei Han, An attribute-oriented rough set approach for knowledge discovery in databases, in: Rough Sets, Fuzzy Sets and Knowledge Discovery: Proceedings of the International Workshop on Rough Sets and Knowledge Discovery (RSKD'93), Banff, Alberta, Canada, 12–15 October 1993, Springer London, 1994.
[66] Xiaohua Hu, Nick Cercone, Learning in relational databases: a rough set approach, Comput. Intell. 11 (2) (1995) 323–338.
[67] Guo-yin Wang, Yu Hong, D.C. Yang, Decision table reduction based on conditional information entropy, Chinese Journal of Computers-Chinese Edition-25 (7) (2002) 759–766.
[68] Guo-Yin Wang, et al., Theoretical study on attribute reduction of rough set theory: comparison of algebra and information views, in: Proceedings of the Third IEEE International Conference on Cognitive Informatics, 2004, IEEE, 2004.
[69] Haleh Vafaie, Ibrahim F. Imam, Feature selection methods: genetic algorithms vs. greedy-like search, Proceedings of the international conference on fuzzy and intelligent control systems 51 (1994).
[70] Anders Torvill Bjorvand, Komorowski Jan, Practical applications of genetic algorithms for efficient reduct computation, Wissenschaft & Technik Verlag 4 (1997) 601–606.
[71] Emrah Hancer, New filter approaches for feature selection using differential evolution and fuzzy rough set theory, Neural Comput. Appl. 32 (2020) 2929–2944.
[72] Xiangyang Wang, et al., Feature selection based on rough sets and particle swarm optimization, Pattern Recogn. Lett. 28 (4) (2007) 459–471.
[73] Richard Jensen, Qiang Shen, Finding rough set reducts with ant colony optimization, Proceedings of the 2003 UK workshop on computational intelligence 1 (2) (2003).
[74] Mohamed A. Tawhid, M. Ibrahim Abdelmonem, Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm, International journal of machine learning and cybernetics 11 (2020) 573–602.
[75] W.Y. Tang, K.Z. Mao, Feature selection algorithm for data with both nominal and continuous features, in: T.B. Ho, D. Cheung, H. Liu (Eds.), PAKDD 2005, Lecture Notes in Artificial Intelligence vol. 3518, Springer, Berlin, Heidelberg, 2005, pp. 683–688.
[76] W. Pedrycz, G. Vukovich, Feature analysis through information granulation and fuzzy sets, Pattern Recogn. 35 (2002) 825–834.
[77] Seyedali Mirjalili, Andrew Lewis, The whale optimization algorithm, Adv. Eng. Software 95 (2016) 51–67.
[78] Majdi Mafarja, Seyedali Mirjalili, Whale optimization approaches for wrapper feature selection, Appl. Soft Comput. 62 (2018) 441–453.
[79] S. Jadhav, H.M. He, K. Jenkins, Information gain directed genetic algorithm wrapper feature selection for credit rating, Appl. Soft Comput. 69 (AUG 2018) 541–553, https://doi.org/10.1016/j.asoc.2018.04.033.
[80] T. F, R. R, A. R, Holdout validation for comparison classfication naïve bayes and KNN of recipient kartu Indonesia pintar, IOP Conf. Ser. Mater. Sci. Eng. 1125 (1) (2021), 012041.
[81] J. HoKun, Y. ChanSu, Enhancement of ship type classification from a combination of CNN and KNN, Electronics 10 (10) (2021) 1169.
[82] F. Zhu, X. Jia-kun, W. Zhong-yu, L. Pei-Chen, Q. Shu-jun, H. Lei, Image classification method based on improved KNN algorithm, J. Phys. Conf. 1930 (1) (2021).
[83] Tarun Maini, et al., Rough set based feature selection using swarm intelligence with distributed sampled initialisation, in: 2017 6th International Conference on Computer Applications in Electrical Engineering-Recent Advances (CERA), IEEE, 2017.
[84] Jayashree Piri, Puspanjali Mohapatra, An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection, Comput. Biol. Med. 135 (2021), 104558.

[85] Jiao Hu, et al., Orthogonal learning covariance matrix for defects of grey wolf optimizer: insights, balance, diversity, and feature selection, Knowl. Base Syst. 213 (2021), 106684.
[86] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381, https://doi.org/10.1016/j.neucom.2015.06.083.
[87] C. Qin et al., "Anti-noise diesel engine misfire diagnosis using a multi-scale CNN-LSTM neural network with denoising module," CAAI Transactions on Intelligence Technology, vol. n/a, no. n/a, doi: https://doi.org/10.1049/cit2.12170..
[88] K. Zhao, Z. Jia, F. Jia, H. Shao, Multi-scale integrated deep self-attention network for predicting remaining useful life of aero-engine, Eng. Appl. Artif. Intell. 120 (2023), 105860.
[89] Z. Peng, et al., A novel optimal bipartite consensus control scheme for unknown multi-agent systems via model-free reinforcement learning, Appl. Math. Comput. 369 (2020), 124821.
[90] Kenneth V. Price, Differential evolution, in: Handbook of Optimization: from Classical to Modern Approach, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 187–214.
[91] James Kennedy, Eberhart Russell, Particle swarm optimization, in: Proceedings of ICNN'95-international Conference on Neural Networks vol. 4, IEEE, 1995.
[92] Sanjoy Chakraborty, et al., COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction, Comput. Biol. Med. 139 (2021), 104984.
[93] Jie Xing, et al., Boosting whale optimizer with quasi-oppositional learning and Gaussian barebone for feature selection and COVID-19 image segmentation, Journal of bionic engineering 20 (2) (2023) 797–818.
[94] Heikki Maaranen, Kaisa Miettinen, Marko M. Mäkelä, Quasi-random initial population for genetic algorithms, Comput. Math. Appl. 47 (12) (2004) 1885–1895.
[95] Chih-Hsun Chou, Jou-Nan Chen, Genetic algorithms: initialization schemes and genes extraction, in: Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No. 00CH37063) vol. 2, IEEE, 2000.
[96] Heikki Maaranen, Kaisa Miettinen, Antti Penttinen, On initial populations of a genetic algorithm for continuous optimization problems, J. Global Optim. 37 (2007) 405–436.
[97] Y. Xu, H. Chen, A.A. Heidari, J. Luo, Q. Zhang, X. Zhao, C. Li, An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks, Expert Syst. Appl. 129 (2019) 135–155.
[98] H. Ren, J. Li, H. Chen, C. Li, Adaptive levy-assisted salp swarm algorithm: analysis and optimization case studies, Math. Comput. Simulat. 181 (2021) 380–409.
[99] P. Civicioglu, E. Besdok, M.A. Gunen, U.H. Atasever, Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms, Neural Comput. Appl. 32 (2020) 3923–3937.
[100] W.-N. Chen, et al., Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2) (April 2013) 241–258, https://doi.org/10.1109/TEVC.2011.2173577.
[101] J. Hu, W. Gui, A.A. Heidari, Z. Cai, G. Liang, H. Chen, Z. Pan, Dispersed foraging slime mould algorithm: continuous and binary variants for global optimization and wrapper-based feature selection, Knowl. Base Syst. 237 (2022), 107761.
[102] X. Chen, K. Yu, W. Du, W. Zhao, G. Liu, Parameters identification of solar cell models using generalized oppositional teaching learning based optimization, Energy 99 (2016) 170e80.
[103] Shuyin Xia, et al., GBNRS: a novel rough set algorithm for fast adaptive attribute reduction in classification, IEEE Trans. Knowl. Data Eng. 34 (3) (2020) 1231–1242.
[104] Yuhua Qian, et al., Fuzzy-rough feature selection accelerator, Fuzzy Set Syst. 258 (2015) 61–78.
[105] Y. Han, et al., Multi-verse optimizer with rosenbrock and diffusion mechanisms for multilevel threshold image segmentation from COVID-19 chest X-ray images, JBE 20 (3) (2023) 1198–1262.
[106] L. Ren, et al., Gaussian kernel probability-driven slime mould algorithm with new movement mechanism for multi-level image segmentation, Measurement 192 (2022), 110884.
[107] X. Wang, et al., Crisscross Harris hawks optimizer for global tasks and feature selection, JBE 20 (3) (2023) 1153–1174.
[108] C. Lin, et al., Double mutational salp swarm algorithm: from optimal performance design to analysis, JBE 20 (1) (2023) 184–211.
[109] L. Hu, et al., An intelligent prognostic system for analyzing patients with paraquat poisoning using arterial blood gas indexes, J. Pharmacol. Toxicol. Methods 84 (2017) 78–85.
[110] X. Yu, et al., Synergizing the enhanced RIME with fuzzy K-nearest neighbor for diagnose of pulmonary hypertension, Comput. Biol. Med. 165 (2023), 107408.
[111] J. Xia, et al., Adaptive barebones salp swarm algorithm with quasi-oppositional learning for medical diagnosis systems: a comprehensive analysis, JBE 19 (1) (2022) 240–256.
[112] J. Xia, et al., Generalized oppositional moth flame optimization with crossover strategy: an approach for medical diagnosis, JBE 18 (4) (2021) 991–1010.
[113] H. Zhang, et al., Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems, Eng. Comput. 39 (3) (2023) 1735–1769.
[114] H. Chen, et al., Slime mould algorithm: a comprehensive review of recent variants and applications, Int. J. Syst. Sci. (2022) 1–32.