Neural Processing Letters (2024) 56:110
https://doi.org/10.1007/s11063-024-11553-9

A Feature Selection Method Based on Feature-Label Correlation Information and Self-Adaptive MOPSO

Fei Han[1,2] · Fanyu Li[1,2] · Qinghua Ling[3] · Henry Han[4] · Tianyi Lu[1] · Zijian Jiao[1] · Haonan Zhang[1]

Accepted: 10 February 2024

Abstract
Feature selection can be seen as a multi-objective task, where the goal is to select a subset of features that exhibit minimal correlation among themselves while maximizing their correlation with the target label. Multi-objective particle swarm optimization algorithm (MOPSO) has been extensively utilized for feature selection and has achieved good performance. However, most MOPSO-based feature selection methods are random and lack knowledge guidance in the initialization process, ignoring certain valuable prior information in the feature data, which may lead to the generated initial population being far from the true Pareto front (PF) and influence the population's rate of convergence. Additionally, MOPSO has a propensity to become stuck in local optima during the later iterations. In this paper, a novel feature selection method (fMOPSO-FS) is proposed. Firstly, with the aim of improving the initial solution quality and fostering the interpretability of the selected features, a novel initialization strategy that incorporates prior information during the initialization process of the particle swarm is proposed. Furthermore, an adaptive hybrid mutation strategy is proposed to avoid the particle swarm from getting stuck in local optima and to further leverage prior information. The experimental results demonstrate the superior performance of the proposed algorithm compared to the comparison algorithms. It yields a superior feature subset on nine UCI benchmark datasets and six gene expression profile datasets.

Keywords Feature selection · Multi-objective optimization · Particle swarm optimization · Mutual information · Self-adaptation

✉ Fei Han
hanfei@ujs.edu.cn

1 School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, Jiangsu, China

2 Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Zhenjiang 212013, Jiangsu, China

3 School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang 212100, China

4 School of Engineering and Computer Science, Baylor University, Waco 76798, USA

Published online: 18 March 2024

1 Introduction

In this era of data explosion, as the number of instances and the dimensionality of data continue to increase, the processing and parsing of data have become increasingly challenging. Feature selection (FS) is a mainstream data reduction technique that aims to eliminate redundant and noisy attributes. Its primary objective is to select the smallest subset of features from the original feature set based on a FS criterion [1]. The advantage of FS lies in its ability to compress the search space of the learning algorithm and lower the size of the feature set, thereby diminishing the dimensionality of the data, easing the learning task, and improving model efficiency [2].

In recent years, numerous researchers have applied swarm intelligence evolutionary algorithms (EA) to the field of FS. Swarm intelligence optimization algorithms exhibit characteristics such as simple operation, fast convergence, and robust global search ability, making them well-suited for tackling intricate optimization problems. Swarm intelligence algorithms, including genetic algorithm (GA) [3, 4], artificial bee swarm algorithm (ABO) [5], grey wolf algorithm (GWO) [6], particle swarm algorithm (PSO) [7], have demonstrated promising results. Among them, PSO stands out as one of the most frequently employed optimization techniques. PSO is not only used to feature selection problems, but also widely applied in other fields. Many scholars have made different improvements to in different fields. Wang et al. [8] proposed a particle swarm optimization algorithm based on reinforcement learning level (RLLPSO) for large-scale problems, which increases the diversity of the population, improves the search performance and convergence speed of the population. Inspired by conditional integrals in automatic control, Xiang et al. [9] proposed an adaptive search direction learning method for PSO (ISPSO). This method has faster global convergence speed and higher solution accuracy. Xia et al. [10] proposed an MFCPSO algorithm to address the shortcomings of fitness based selection, which exhibits promising characteristics in large-scale complex functions. However, these evolutionary optimization algorithms also have certain limitations. Most of them are designed for single-objective FS problems, whereas FS can be viewed as a multi-objective optimization problem. Typically, two optimization objectives are considered: maximizing the classification accuracy of the selected feature subset and minimizing the size of the subset. In fact, researchers have explored the use of multi-objective EA, including the MOPSO algorithm, for solving the FS problem. Pradip et al. [11] proposed a two-phase multi-objective FS method aimed at selecting the most relevant features. The one phase involves global search using PSO, while in the other phase, a combination of PSO and GWO, based on a modified Newton's second law of motion, performs local search starting from the results obtained in the global search. Wang et al. [12] introduced a multi-objective evolutionary FS algorithm that incorporates a correlation metric and a novel redundancy metric for class correlation redundancy. The method uses Pareto optimality to assess a subset of candidate features to find the compact feature subset with maximum correlation and minimum redundancy. Xue et al. [13] proposed a FS adaptive multi-objective genetic algorithm, which incorporates an adaptive mechanism to dynamically select five different crossover operators during various evolutionary processes, allowing the algorithm to remove multiple features while ensuring classification performance. Feng et al. [14], aims to improve the global search capability and mitigate the stagnation of local optimal solutions phenomenon, the model was modified in the PSO part using genetic operators and Levy flight. These algorithms strive to discover a collection of solutions that strike a balance between classification precision and the size of the selected feature subset.

However, these algorithms ignore the prior information contained in the feature data during the initialization process, and use random initialization methods to generate initial solutions that may be far from the true Pareto front, affecting the convergence speed of the population. To alleviate this problem, Han et al. [15] introduced an improved feature selection method, which sets the selection threshold according to the correlation between features and categories in the initialization stage to select feature subsets of superior quality. Yu et al. [16] presented a swarm initialization strategy that combines blended initialization and threshold selection techniques. Additionally, PCA is employed to rank the importance of features. Although these methods take into account the prior information contained in the feature data during the initialization process, the particles are still susceptible to fall into local optima. Aiming at this problem, Fu et al. [17] introduced a novel multi-objective binary GWO method that incorporates a guided mutation strategy. The method utilizes the Pearson correlation coefficient to guide local search, enhancing the population's ability to explore local regions. Additionally, a dynamic perturbation mechanism is employed for mutation, preventing population stagnation caused by a single strategy. This dynamic adjustment ensures population diversity is maintained and improves the algorithm's detection capability. Zhou et al. [18] presented an adaptive hierarchical update PSO algorithm to overcome the issue of particle swarm algorithms frequently getting trapped in local optima and struggling to escape. The proposed method incorporates multi-level update formulas for both the global exploration subgroup and the local exploitation subgroup. This approach enhances the resistance to local optima and improves the algorithm's ability to explore globally optimal solutions. Wei et al. [19] employed a neighborhood search strategy to enhance the local search capability of the swarm during stagnation periods. Xiang et al. [20] proposed a PID based PSO strategy (PBS-PSO) to avoid premature convergence of particle swarm optimization, in order to accelerate convergence and adjust the search direction to escape local optima. Xue et al. [13] introduced a mechanism for detecting search stagnation aimed at mitigating premature convergence in PSO. Although these methods can avoid particle swarms from getting trapped in local optima, due to most of them are lack of prior information guidance, restrict the search performance of swarm intelligence algorithms and hinders their ability to converge towards the global optimum.

Based on the above analysis, incorporating prior knowledge into both the population initialization and search process would inevitably expedite the algorithm's search speed and enhance the explainability of the selected features. Introducing prior information in the initialization process can bring the generated initial solutions closer to the true Pareto front, accelerate population convergence speed, and also increase the diversity of population particles. Coupling prior knowledge into the search process can effectively guide particles to search in a better direction, improve the search performance of the population. Therefore, this paper proposes an adaptive multi-objective particle swarm feature selection algorithm based on feature-label relevance information guidance, combining the advantages of filtered and wrapped FS algorithms on the basis of full consideration of prior information. The primary differentiating factors of this paper from other algorithms can be summarized as follows:

Firstly, a strategy for setting feature encoding intervals is proposed, which determines the interval boundaries based on the magnitude of correlation between features and categories. This strategy increases the probability of selecting features with higher correlation to the categories, thereby enhancing the explainability of the selected features.

Secondly, a novel swarm initialization method based on feature-label correlation is proposed. This method improves the quality of initial solutions and the distribution of particles, resulting a significant improvement in the proximity of initial solutions to the true Pareto front. Additionally, It expedites the rate at which the population converges.

Finally, an adaptive hybrid perturbation strategy is proposed to facilitate the particles in escaping from local optima, taking into account the performance of the particles, the selection probability of the features and the selection situation.

The paper is organized as follows in the subsequent sections: Sect. 2 presents an overview of existing work related to MOPSO, and information entropy. Sect. 3 provides the proposed FS algorithm. In Sect. 4, the experimental results are presented and analyzed, providing a comprehensive discussion of the obtained findings. Finally, Sect. 5 gives the conclusions of this paper.

2 Preliminaries

2.1 Multi-objective Optimization Problems (MOPs)

Problems with multiple optimization objectives are called multi-objective problems, and since the objectives are in conflict with each other, a solution cannot be optimal for all objectives. The solutions that satisfy the Pareto optimality criteria in such problems are referred to as Pareto optimal solutions. These solutions allow for a trade-off among different objective functions, as improving one objective may come at the expense of another [21, 22]. The minimum MOP can be described in the following manner:

$$\begin{aligned} minimize\ F(x) &= (f_1(x), f_2(x), \ldots, f_n(x)) \\ subject\ to: u_i(x) &\leq 0, i = 1, 2, \ldots, k \\ e_j(x) &= 0, j = 1, 2, \ldots, k \end{aligned} \tag{1}$$

where $X = (x_1, x_2, x_3, \ldots, x_D)$ represents the D-dimensional vector in decision space and n is the number of objectives, $f_i(X)$ indicates the ith minimized objective function, $u_i(X)$ and $e_j(X)$are the inequality and equlaity constraints, respectively. Given two feasible solutions X_1 and X_2, X_1 dominates X_2, if and only if for $\forall_a,\ f_a(X_1) \leq f_a(X_2)$ and $\exists b,\ f_b(X_1) < f_b(X_2),\ a, b \in \{1, 2, \ldots, n\}$. If no other solution dominates X^*, then X^* is known as a Pareto-optimal solution. The set of all Pareto-optimal solutions is known as the Pareto-optimal set, while the objective values associated with these solutions form the Pareto front.

2.2 Particle Swarm Optimization

PSO has been widely used in a diverse range of optimization problems [23, 24]. In the particle swarm algorithm, each particle corresponds to a prospective solution to an optimization problem, and collectively, all particles form a set of candidate solutions. Each particle possesses two fundamental properties: velocity and position. The update of velocity and position for the particle swarm is performed as follows.

$$v_i(t+1) = \omega * v_i(t) + c_1 * r_1 * (pbest_i - x_i(t)) + c_2 * r_2 * (gbest_i - x_i(t)) \tag{2}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), i = 1, 2, \ldots, n \tag{3}$$

where ω represents the inertia weight, t represents the number of current iterations, c_1 and c_2 are the learning factors, r_1 and r_2 are two random values uniformly distributed in the interval [0,1], and $pbest_i$ and $gbest_i$ serve as representations for the individual optimal position and the global optimal position, respectively, of particle i.

2.3 Information Entropy

2.3.1 Entropy

Entropy quantifies the level of uncertainty associated with a random variable. Higher entropy corresponds to greater uncertainty in the random variable. The entropy of a continuous random variable X, denoted as H(X), is defined by the following equation:

$$H(X) = -\sum_{x \in X} p(x) log(p(x)) \tag{4}$$

where X denotes the random variable and $p(x)$ is the probability density function of X.

2.3.2 Relative Entropy

Relative entropy is a measure that quantifies the difference or dissimilarity between two probability distributions. It provides a measure of how one distribution differs from another in terms of their information content or structure. Specifically, it measures the additional amount of information needed to encode data from one distribution using a code optimized for another distribution. The definition of the relative entropy between probability distributions $p(x)$ and $q(x)$ is as follows:

$$D(p||q) = \sum_{x \in X} p(x) log \frac{p(x)}{q(x)} \tag{5}$$

2.3.3 Mutual Information (MI)

MI is a measure used to quantify the amount of information that one random variable contains about another random variable [25]. It reflects the degree of correlation between the variables, with higher values indicating stronger correlation. The MI between two discrete variables X and Y is defined as follows:

$$(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) log \frac{p(x, y)}{p(x)p(y)} = D(p(x, y)||p(x)p(y)) \tag{6}$$

where $p(x, y)$ denotes the joint probability density of x and y, and $p(x)$ and $p(y)$ refer to the marginal probability densities of x and y respectively.

The relationship between MI and entropy can be described as follows:

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \tag{7}$$

3 The Proposed Method

In this section, in order to improve the quality of the initial solutions of the population and to expedite the convergence process. A novel particle swarm initialization strategy is proposed, which couples prior information in the initialization process and enhances the explainability of the selected features. At the same time, an adaptive hybrid perturbation strategy is proposed in order to avoid the PSO algorithm from falling into local optimal solutions. The specific details of the two strategies are as follows.

3.1 A Novel Initialization Strategy

To enhance the dispersion of particles and improve the quality of initial solutions, it is essential to thoroughly take into account the interrelation between features and categories. In this paper, mutual information is utilized as a metric to assess the correlation between features and labels. A higher value of mutual information indicates a stronger relevance between the features and labels. In order to ensure the diversity of particles, half of the particles of the population are initialized using feature-label guidance, while the other half is left to be initialized randomly. The overall process is illustrated in Algorithm 1.

Algorithm 1 The proposed initialization strategy

Input: Dataset, D(the number of features),N(Number of Particles)
Output: NewPopulation
1: Calculate the MI of value of each feature using Eq. (6);
2: **for** k = 1 to D **do**
3: Calculate feature coding intervals using Eq. (8)and Eq. (9);
4: **end for**
5: **for** i=1 to $N/2$ **do**
6: $P_i \leftarrow$ Initialize particles based on feature-label correlation information
7: **end for**
8: **for** i=$N/2$ to N **do**
9: $P_i \leftarrow$ Randomly initialized
10: **end for**
11: **return** NewPopulation

3.1.1 The Initialization Strategy Based on Feature-Label Correlation Information

FS can be viewed as a binary optimization problem since it entails making decisions on whether to select or exclude features. While binary PSO can directly encode particle positions as binary values, continuous PSO has shown better performance in FS [26]. Therefore, in this paper, continuous PSO is employed to adjust the position information of particles in the FS algorithm. Nonetheless, evaluating fitness in continuous particle swarm algorithms is challenging, requiring the conversion of real values to binary values before fitness evaluation. In the conversion process, most PSO-based feature selection algorithms encode particle position information in the range of [0, 1] and use a fixed conversion threshold. However, this fixed feature encoding interval and conversion threshold do not adequately incorporate the correlation information between features and categories. To tackle this problem, we propose a feature encoding interval setting strategy based on feature-label correlation.

Different feature coding intervals are set according to the magnitude of the correlation value. This paper divides the encoding interval of features into two categories: one sets the lower bound of the feature encoding interval (X_{lb}), and the other sets the upper bound of the feature encoding interval (X_{ub}). The rules for setting the interval bounds are as follows.

X_{lb} is set when the correlation value between features and categories exceeds the average correlation value across all features and labels. Conversely, X_{ub} is set when the correlation value is below the average. The calculation formulas are shown in Eq.(8) and Eq.(9).

$$X_{lb} = \alpha * \frac{I(f_j, C)}{max(I(f, C))}, \ j = 1, 2, ..., D, \alpha = 0.2 \tag{8}$$

$$X_{ub} = T + \beta * \frac{I(f_j, C)}{max(I(f, C))}, j = 1, 2, \ldots D, \beta = 0.4 \tag{9}$$

where $I(f_j, C)$ represents the value of the MI between the feature and the category C. T represents for selection threshold. α,β are two different moderators, the exact values of which are discussed in Sect. 4.6.

The encoding process of the features is as follows. Taking a data set with D-dimensional features as an example, then the position information of the ith particle can be represented by a string of D-dimensional real-valued data, denoted as vector $F_i = (x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,D})$. The range of values of each component in F_i is divided into two cases, as shown in Eq.(10)

$$x_{i,j} \in \begin{cases} [X_{lb}, 1], I(f_j, C) > MeanMl \\ [0, X_{ub}], I(f_j, C) \leq MeanMl \end{cases}, \quad i = 1, 2, 3, \ldots, N, j = 1, 2, 3, \ldots, D \tag{10}$$

where $MeanMl$ represents the mean of all feature-label relevance values.

Based on the equation mentioned above, it can be inferred that, given a fixed selection threshold, a higher mutual information value leads to a wider interval of selected features. This ensures that features with a stronger relevance have a higher likelihood of being chosen for selection.

The random initialization of particle position information is shown in Eq.(11).

$$x_{i,j} \in [0, 1], i = 1, 2, 3, \ldots, N, j = 1, 2, 3, \ldots, D \tag{11}$$

Similar to the approach used in HMPSOFS [27], this paper utilizes a consistent binary threshold. Consequently, the particle's position is converted into a binary value for each dimension, relying on this threshold. The conversion process is shown in Eq. (12), $x_{i,j}$ is set to 1 when $x_{i,j}$ is greater than T, otherwise it is set to 0.

$$F_{i,j} = \begin{cases} 1, & x_{i,j} > T \\ 0, & x_{i,j} \leq T \end{cases} \tag{12}$$

where $F_{i,j}$ denotes the jth feature belonging to the feature subset F_i. When $F_{i,j} = 1$ represents that the feature is selected and $F_{i,j} = 0$ means that the feature is not selected. Referring to previous studies [13, 22, 23], the threshold T is set to 0.6 in this paper.

3.2 The Adaptive Hybrid Mutation Strategy

To leverage the correlation information between features and labels more effectively and prevent the particle swarm from converging to local optima, an adaptive hybrid mutation strategy is introduced. The age threshold in dMOPSO [28] is introduced to determine whether the particles fall into a local optimum. In the early stage of the algorithm operation, the particle swarm exhibits powerful search ability and the individual optimal positions of the particles are updated continuously during the search process. However, with the progression of population updates, the search ability of the particles gradually declines, resulting in the particles easily entering a stagnant state. When the age of the particle is below the predetermined, it indicates that the particle still possesses good search ability, so the particle is slightly perturbed by using non-uniform mutation [29]. On the other hand, When the particle's age goes beyond the preset age limit, it implies that the particle is likely to trapped in a local optimum and requires a larger perturbation, so adaptive variation approach is implemented to support the particle in

breaking free from the local optimum and exploring different domains of the solution space. The detailed process is outlined in Algorithm 2.

Algorithm 2 The adaptive hybrid mutation strategy

Input: Pop(swarm), Pbest, N(Number of particles), Ta(threshold of age)
Output: NewPopulation
1: Calculate the probability of mutation using Eq.(17) and Eq.(18)
2: **for** i=1 to N **do**
3: **if** age(P_i) <= Ta **then**
4: $P_i \leftarrow$ Nonuniform Mutation(P_i);//According to section 3.2.1
5: age(P_i) = age(P_i) + 1;
6: **else**
7: $P_i \leftarrow$ Adapative Mutation(P_i);//According to section 3.2.2
8: age(P_i) = 0;
9: **end if**
10: **end for**
11: **return** NewPopulation

3.2.1 Non-uniform Mutation

The non-uniform mutation operator φ incorporates a dynamic decrease in mutation probability as the number of iterations increases. During the iteration process, the PSO algorithm has been pursuing the balance between exploration and exploitation. In the early stage of the iteration, by increasing the exploration intensity, the algorithm is more likely to find the global optimal solution or a solution close to the optimal solution. Therefore, using a higher mutation probability can improve the global search ability of particles. In the later stages of the iteration, when the search space is reduced and the global optimal solution is closer, local search becomes more important. At this time, the mutation probability is reduced and the exploitation of existing excellent solutions is increased.

$$x_{i,j} = \begin{cases} x_{i,j} + Pbest_{i,j} * (1 - r^{(\varphi)^{\lambda}}), r \leq 0.5 \\ x_{i,j} - Pbest_{i,j} * (1 - r^{(\varphi)^{\lambda}}), r > 0.5 \end{cases} \tag{13}$$

$$\varphi = 1 - \frac{t}{maxIt} \tag{14}$$

where r is a random number in the range of 0 to 1, t is the current number of iterations of the population, and $maxIt$ is the maximum number of iterations. λ is a system parameter that determines the dependence of the random number perturbation on the number of iterations, and based on related research [30, 31], the algorithm proposed in this chapter λ will be set to 3.

3.2.2 Adpative Mutation

The adaptive mutation strategy further utilizes the prior information contained in the feature data, and calculates the feature mutation probability according to the performance of the particle itself, combined with the selection probability of the feature and whether the feature is selected.

Firstly, the performance of the particle is defined. Without considering any preferences, the Euclidean distance between the particle's position in the target space and the origin of

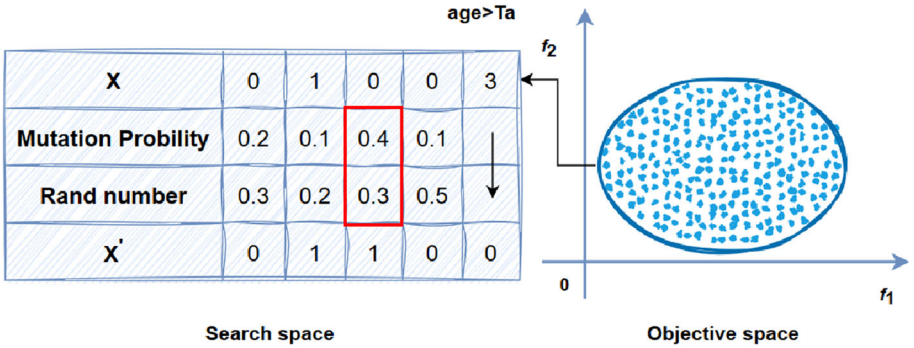

Fig. 1 An example of a specific perturbation

the target space is employed as a metric to evaluate the particle's performance. A smaller distance indicates better performance for the particle. It is calculated as follows:

$$performance_i = \left\| \overline{f(x)} \right\|_2 \tag{15}$$

where $performance_i$ denotes the performance of the ith particle, denotes the target vector of particle i.

Next, the probability of a feature being selected is calculated based on the feature encoding interval set during initialization. The feature coding interval length is $1 - X_{lb}$ or X_{ub} and a fixed selection threshold T is used. The selection probability of a feature (P_s) is defined as follows:

$$P_s = \begin{cases} \dfrac{1-T}{1-X_{lb}}, x_{i,j} \epsilon [X_{lb}, 1] \\ \dfrac{X_{ub}-T}{X_{ub}}, x_{i,j} \epsilon [0, X_{ub}] \end{cases} \tag{16}$$

For a feature f_j, as shown in Eq.(16), different feature encoding intervals correspond to different mutation probabilities.

Finally, the probability of variation is calculated based on the selection of features, which is divided into the following two cases.

Case 1: If the feature is selected, its mutation probability will be calculated as follows:

$$MP = exp(-P_S) * (1 - performance_i) \tag{17}$$

Case 2: If the feature is not selected, the probability of its mutation will be calculated as follows:

$$MP = (1 - exp(-P_S)) * (1 - performance_i) \tag{18}$$

An example of a specific perturbation is shown in the Fig. 1. When the MP of feature exceeds the generated random number, the mutation operation is executed, and vice versa.

3.3 The Framework of the Proposed Method

Algorithm 3 outlines the general framework of fMOPSO-FS. The fMOPSO-FS framework primarily comprises two phases. The first stage is the initialization stage. Firstly, the mixed

initialization method is used to initialize the population. As outlined in Algorithm 1, certain particles are initialized based on prior information derived from the feature data, while the remaining particles are initialized randomly. Additionally, the external archive and particle ages are also initialized. The main loop stage constitutes the second phase of the framework, which mainly involves the evaluation of particles and the update of adaptive hybrid mutation and external archives. As depicted in Algorithm 2, when the age of the particle exceeds the predefined age threshold, adaptive mutation is performed; otherwise, non-uniform mutation is utilized. Adaptive mixed perturbation strategies can help them break through local optima and increase population diversity. As the external archive is continuously updated, the final set of leader archive serves as the final outcome. In this paper, minimizing the feature subset size and minimizing the classification error rate are chosen as the evaluation functions, which are conflicting objectives. Minimizing the feature subset size is denoted as f_1, minimizing the classification error rate is denoted as f_2. These two evaluation functions are calculated according to the Eq.(19) and Eq.(20) respectively.

$$f_1 = \frac{s_i}{D}, S_i = \sum_{j=1}^{D} F_{i,j} \tag{19}$$

where S_i represents the number of features in the feature subset F_i and D represents the total count of features in the dataset.

$$f_2 = \frac{(FP + FN)}{(FP + FN + TP + TN)} \tag{20}$$

where FP, FN, TP and TN represent false positive, false negative, true positive and true negative respectively.

Algorithm 3 Framework of fMOPSO-FS

Input: Dataset, population size(N), maximal generation number (maxgen)
Output: Archive(A)
1: /*Initialization*/
2: $Pop \leftarrow$ Initialize particles(N); //Initialize Particles according to the Algorithm 1
3: $Age \leftarrow$ Initialize Age(N);
4: $A \leftarrow$ Update Archive(Pop);
5: /*Main loop*/
6: **while** the termination criterion is not fulfilled **do**
7: $Pop, Age \leftarrow$ Adaptive hybrid mutation(Pop,age); //According to the Algorithm 2
8: $A \leftarrow$ Update Archive(A $\cup$ Pop);
9: **end while**
10: **return** NewPopulation

3.4 Computational Complexity Analysis

The proposed algorithm mainly includes two stages: initialization and main loop. The initialization phase mainly includes initializing the velocity and position of particles, as well as external archiving, etc. The main loop phase includes the search process of particles and the selection of global optimal particles, etc. The main time cost in the initialization stage is to calculate the correlation between features and labels, with a time complexity of $O(D + N)$, where D represents the dimension of features, N denote the number of particles. The time

complexity in the main loop stage is mainly affected by the search and update process of particles, and the main time consumption in the particle search process is to calculate the mutation probability of features, with a time complexity of $O(N^2 + D)$, In the process of particle update, the time complexity mainly depends on the selection of leading particles. The selected time complexity is $O(N)$. If the selection and update of particles are carried out serially, the time complexity of the main loop stage is $O(N^2 + D + N)$. Due to N^2 being much larger than N and D, the time complexity of the proposed algorithm is $O(N^2)$. Compared with other similar feature selection algorithms based on particle swarm optimization, although the proposed algorithm increases the calculation of the correlation between features and labels and the probability of feature mutation, the increased time complexity is constant level, so the overall time complexity did not increase.

4 Experiments and Discussion

4.1 Methods of Comparison and Corresponding Parameter Configurations

In this section, we have selected a series of multi-objective FS algorithms for comparison with fMOPSO-FS, which contains several classical and state-of-the-art multi-objective optimization algorithms. The classical multi-objective optimization algorithms consist of MOPSO [32], NSGAIII [33], MOEA/D [34], and the advanced multi-objective optimization algorithms encompass HMPSOFS [27], RFPSOFS [35], MOEA/D-COPSO [36], and AGMOPSO [15]. All four of these advanced algorithms employ the PSO to discover optimal solutions.

To guarantee the impartiality of the comparative experiment, regarding the dataset processing, to begin the experiment, the dataset undergoes a random partitioning process, where it is divided into two subsets. The training set comprises 70% of the data, while the remaining 30% is designated as the test set. Additionally, a 10-fold cross validation technique is employed to evaluate the model. This approach helps mitigate the risk of overfitting the model on the training set and enhances the reliability of the training process. The classification error rate of each particle is computed using the K Nearest Neighbor (KNN) classifier, with k set to 5. The settings of parameters for different algorithms are presented in Table 1. These algorithms are implemented on MATLAB R2020b, Intel(R) Core(TM) i5-8265U, 1.80 GHz, 8GB RAM.

4.2 Performance Metrics

To gauge the effectiveness of the comparison algorithm and the fMOPSO-FS algorithm, two commonly used metrics, namely hypervolume (HV) and inverted generational distance (IGD), are employed. These metrics, HV and IGD, are considered as the most representative measures for evaluating the performance of optimization algorithms.

The evaluation method for the HV metric was initially introduced by Zitzler et al. [37]. The diversity and convergence of an algorithm are assessed by measuring the volume of the hypercube formed by the individuals in the Pareto solution set and the reference points in the target space. The larger the HV value is, the better the Pareto front set is. This evaluation method quantifies the spread and performance of the algorithm's solutions. In the specific research paper mentioned, the reference point is defined as (1.0, 1.0) based on the objective function's design. The formula to calculate the HV is as follows:

Table 1 Parameter configurations for seven algorithms

Algorithm	Private parameters	Common parameters
MOPSO	The number of grids is fixed at 30 and β is set to 10	Population size $N = 30$
	Crossover probability $proC = 1$	The number of iterations $maxIt = 100$
NSGAIII	Crossover distribution $disC = 20$;	Acceleration rate $c_1 = c_2 = 1.46$
	Mutation probability $proM = 1$	Inertia weight $\omega = 0.729$
	Mutation distribution $disM = 20$	Maximum velocity $v_{\max} = 0.6$
MOEA/D	$T = N/10$	
HMPSOFS	Jumping probability $JP = 0.01$	
RFPSOFS	The number of grids is set to 10 and β is set to 2	
MOEA/D-COPSO	The strict particles identified through the Relief-F algorithm	
	make up 35% of the entire particle set and $T = 0.52$.	
AGMOPSO	The reduction factor α is set to 0.7	

$$HV = \delta(\cup_{i=1}^{|S|} v_i) \tag{21}$$

where δ is the Lebesgue measure; $|S|$ denotes the number of non-dominated solutions obtained by the algorithm, and v_i denotes the HV comprising of the reference point and the ith solution in the solution collection.

The IGD is a comprehensive metric for evaluating algorithm performance, and is mainly used to evaluate the convergence performance and distribution performance of the algorithm [38]. A lower IGD value indicates better overall performance of the algorithm in terms of convergence and distribution. However, in multi-objective feature selection problems, there is no true PF available. Therefore, in this paper, the set of non-dominated solutions generated by all compared algorithms and the proposed algorithm in 30 independent runs is considered as the surrogate Pareto front. The calculation of the IGD is performed as follows:

$$IGD(P_s, P^*) = \frac{\Sigma_{x \in P^*} min_{y \in P_s} Dis(x, y)}{|P^*|} \tag{22}$$

where P_s represents the set of Pareto optimal solutions obtained from the algorithm and P^* denotes a collection of uniformly distributed reference points that are sampled from the true PF. $Dis(x, y)$ is the Euclidean distance between point x in P_s and point y in the optimal solution collection obtained by the method.

4.3 Experimental Analysis on UCI Datasets

To evaluate the performance of the fMOPSO-FS, seven UCI datasets are selected as experimental datasets in this subsection. The details of the datasets utilized are outlined and presented in the following Table 2 [39].

Tables 3, 4, 5, 6 show the average and standard deviation of the HV and IGD values obtained by the fMOPSO-FS algorithm and the comparison algorithms on the seven UCI datasets.The values '↑','↓' and 'o' indicate that the comparison algorithm outperforms, underperforms and approximates fMOPSO-FS, respectively, while the values preceding and

Table 2 Details of relevant UCI datasets

Dataset	Number of features	Number of records	Number of classes
German	24	1000	2
Sonar	60	208	2
Hill valley	100	606	2
Musk1	166	476	2
LSVT	310	126	2
Madelon	500	2000	2
Isolet5	617	1559	26
Multiple features	649	2000	15
CNAE	856	1080	9

following the symbol '±' indicates the mean and standard deviation of the relevant algorithm on the dataset, respectively. Since the sample data do not have normality, non-parametric tests are used to compare the differences in the data. Judgment based on the P value obtained by non-parametric test. If the difference in the judgment results is not significant, it means that the performance of the two algorithms is close, represented by '∘'. If there are significant differences, the evaluation will be carried out according to the evaluation methods of different indicators. The larger the HV value, the better the algorithm performance. The smaller the IGD value, the better the algorithm performance. Choose '↑' or '↓' to represent it according to the corresponding situation. The bold font represents the best performing among these algorithms.

When analyzing the results from the training set perspective, as presented in Tables 3 and 4, it is evident that fMOPSO-FS outperforms MOPSO, MOEA/D and NSGAIII, and the obtained HV value and IGD value are close to those of MOEA/D-COPSO and AGMOPSO on the German dataset.The HV value obtained by the MOEA/D-COPSO algorithm is better than that of the fMOPSO-FS algorithm, but its IGD value is worse than that of the fMOPSO-FS algorithm. The stability of fMOPSO-FS shows a slight decrease compared to MOEA/D-COPSO. However, in contrast to other comparison algorithms, fMOPSO-FS consistently outperforms them in terms of HV and IGD values across various datasets.

In accordance with the test set results shown in Tables 5 and 6, it can be seen that fMOPSO-FS achieves HV and IGD values comparable to HMPSOFS, RFPSOFS, MOEA/D-COPSOS, and AGMOPSO on the German dataset. On the Sonar dataset, fMOPSO-FS demonstrates similar performance to HMPSOFS and MOEA/D-COPSOFS. On the Musk1 dataset, fMOPSO-FS exhibits HV and IGD values similar to RFPSOFS and MOEA/D-COPSO, but it outperforms them to emerge as the top-performing algorithm overall. Meanwhile, AGMOPSO obtained better IGD values on the CNAE, but fMOPSO-FS obtained better HV values. MOEA/D-COPSO algorithm has shown good performance in LSVT, with better HV and IGD values than the fMOPSO-FS algorithm and showcases strong stability across the Sonar, Hillvalley, Musk1, and CNAE datasets. In contrast, fMOPSO-FS demonstrates excellent stability specifically on the Sonar and Hillvalley datasets. Nevertheless, fMOPSO-FS does not yield significantly improved results on the German dataset. This could be attributed to the dataset's weak correlation with categories or disregard for potential redundancies among features. Overall, when compared to other comparative algorithms, the fMOPSO-FS algorithm consistently achieves better performance.

Table 3 HV values obtained for each algorithm on the training sets of each dataset

	MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
German	0.7273±	0.7199±	0.6489±	0.7444±	0.7333±	0.7460±	0.7463±	**0.7465**±
	0.0233↓	0.0173↓	0.0396↓	0.0147∘	0.0145↓	0.0099∘	0.0096∘	**0.0075**
Sonar	0.7997±	0.7633±	0.6558±	0.8507±	0.8191±	0.8584±	0.8615±	**0.8794**±
	0.0324↓	0.0298↓	0.0307↓	0.0196↓	0.0405↓	**0.0158**↓	0.0187↓	0.0173
Hill valley	0.5587±	0.5319±	0.4387±	0.5969±	0.5767±	0.6014±	0.5962±	**0.6239**±
	0.0216↓	0.0288↓	0.0243↓	0.0244↓	0.0292↓	0.0210↓	0.0196↓	**0.0144**
Musk1	0.8063±	0.7123±	0.6350±	0.8158±	0.8183±	0.8329±	0.8499±	**0.8627**±
	0.0212↓	0.0245↓	0.0185↓	0.0285↓	0.0463↓	**0.0120**↓	0.0143↓	0.0202
LSVT	0.6524±	0.5333±	0.5978±	0.7672±	0.6765±	**0.8653**±	0.8104±	0.8339±
	0.0479↓	0.0323↓	**0.0187**↓	0.0516↓	0.0517↓	0.0198↑	0.0564∘	0.0190
Madelon	0.6973±	0.5865±	0.4106±	0.8202±	0.7388±	0.5718±	0.8192±	**0.8365**±
	0.0286↓	0.0231↓	**0.0081**↓	0.0220↓	0.0534↓	0.0098↓	0.0155↓	0.0141
Isolet5	0.7645±	0.6213±	0.5476±	0.7449±	0.7747±	0.7106±	0.7782±	**0.7914**±
	0.0147↓	0.0195↓	0.0099↓	0.0183↓	0.0482↓	**0.0100**↓	0.0173↓	0.0138
Multiple	0.8366±	0.7000±	0.6571±	0.8371±	0.8537±	0.8759±	0.8874±	**0.9021**±
Features	0.0216↓	0.0131↓	0.0086↓	0.0194↓	0.0324↓	0.0132↓	0.0136↓	**0.0096**
CNAE	0.7308±	0.5825±	0.5304±	0.7280±	0.7665±	0.7254±	0.8033±	**0.8562**±
	0.0189↓	0.0189↓	0.0164↓	0.0189↓	0.0392↓	0.0109↓	**0.0079**↓	0.0081
↑,↓,∘	**0/9/0**	**0/9/0**	**0/9/0**	**0/8/1**	**0/9/0**	**0/8/1**	**0/7/2**	

Table 4 IGD values obtained for each algorithm on the training sets of each dataset

	MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
German	0.1593±	0.1812±	0.1893±	**0.1325**±	0.1470±	0.1441±	0.1419±	0.1409±
	0.0127↓	0.0193↓	0.0218↓	0.0161↑	**0.0091**↓	0.0102∘	0.0103∘	0.0096
Sonar	0.1600±	0.1822±	0.2823±	0.1275±	0.1529±	0.1266±	0.1217±	**0.1118**±
	0.0223↓	0.0206↓	0.0269↓	0.0162↓	0.0291↓	0.0168↓	0.0149↓	**0.0125**
Hill valley	0.0851±	0.1047±	0.2242±	0.0579±	0.0726±	0.0544±	0.0551±	**0.0429**±
	0.0186↓	0.0254↓	0.0335↓	0.0214∘	0.0219↓	0.0170↓	0.0154↓	**0.0109**
Musk1	0.0745±	0.1607±	0.2413±	0.0798±	0.0759±	0.0648±	0.0489±	**0.0418**±
	0.0150↓	0.0244↓	0.0190↓	0.0252↓	0.0351↓	**0.0081**↓	0.0113∘	0.0152
LSVT	0.2092±	0.2941±	0.3209±	0.1101±	0.1924±	0.0748±	0.0764±	**0.0614**±
	0.0477↓	0.0339↓	0.0161↓	0.0456↓	0.0461↓	**0.0147**↓	0.0606∘	0.0183
Madelon	0.1766±	0.3022±	0.4469±	0.0721±	0.1429±	0.2919±	0.0781±	**0.0688**±
	0.0272↓	0.0210↓	0.0104↓	0.0171∘	0.0505↓	0.0111↓	0.0134↓	**0.0076**
Isolet5	0.0790±	0.2282±	0.2846±	0.1033±	0.0848±	0.1545±	0.0788±	**0.0654**±
	0.0116↓	0.0178↓	0.0083↓	0.0168↓	0.0365↓	0.0115↓	0.0127↓	**0.0099**
Multiple	0.0864±	0.2148±	0.2798±	0.0916±	0.0800±	0.0683±	0.0564±	**0.0510**±
Features	0.0139↓	0.0137↓	0.0092↓	0.0145↓	0.0199↓	0.0090↓	0.0081↓	**0.0050**
CNAE	0.1787±	0.3324±	0.3462±	0.2036±	0.1406±	0.1920±	0.1245±	**0.1026**
	0.0197↓	0.0078↓	0.0102↓	0.0219↓	0.0377↓	**0.0112**↓	0.0167↓	0.0235
↑,↓,∘	**0/9/0**	**0/9/0**	**0/9/0**	**1/6/2**	**0/9/0**	**0/8/1**	**0/6/3**	

Table 5 HV values obtained for each algorithm on the test sets of each dataset

	MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
German	0.7015±	0.7011±	0.6346±	0.7140±	0.7077±	**0.7198**±	0.7153±	0.7143±
	0.0259↓	0.0238↓	0.0390↓	0.0181∘	0.0167∘	**0.0163**∘	0.0176∘	0.0171
Sonar	0.7380±	0.7053±	0.6409±	0.7839±	0.7653±	0.7965±	0.7706±	**0.8014**±
	0.0397↓	0.0470↓	0.0379↓	0.0435∘	0.0523↓	0.0313∘	0.0342↓	**0.0296**
Hill valley	0.5370±	0.5185±	0.4329±	0.5586±	0.5442±	0.5727±	0.5541±	**0.5844**±
	0.0239↓	0.0218↓	0.0240↓	0.0255↓	0.0307↓	0.0256↓	0.0256↓	**0.0207**
Musk1	0.7747±	0.6855±	0.6343±	0.7916±	0.7974±	0.8121±	0.8060±	**0.8219**±
	0.0291↓	0.0323↓	0.0218↓	0.0277↓	0.0490∘	**0.0157**∘	0.0275↓	0.0242
LSVT	0.6144±	0.5121±	0.5797±	0.7079±	0.6291±	**0.8082**±	0.7276±	0.7464±
	0.0621↓	0.0476↓	**0.0289**↓	0.0383↓	0.0692↓	0.0398↑	0.0621∘	0.0349
Madelon	0.6844±	0.5693±	0.4032±	0.8157±	0.7357±	0.5493±	0.7993±	**0.8277**±
	0.0308↓	0.0254↓	**0.0122**↓	0.0232∘	0.0509↓	0.0151↓	0.0164↓	0.0205
Isolet5	0.7474±	0.6037±	0.5410±	0.7307±	0.7641±	0.6977±	0.7573±	**0.7745**±
	0.0181↓	0.0202↓	0.0123↓	0.0244↓	0.0506↓	**0.0127**↓	0.0195↓	0.0168
Multiple	0.8173±	0.6862±	0.6552±	0.8170±	0.8437±	0.8678±	0.8667±	**0.8834**±
Features	0.0236↓	0.0166↓	0.0111↓	0.0208↓	0.0287↓	0.0183↓	**0.0133**↓	0.0168
CNAE	0.7150±	0.5736±	0.5307±	0.7122±	0.7578±	0.7152±	0.8013±	**0.8387**±
	0.0238↓	0.0177↓	0.0199↓	0.0158↓	0.0395↓	0.0185↓	**0.0118**↓	0.0172
↑,↓,∘	**0/9/0**	**0/9/0**	**0/9/0**	**0/6/3**	**0/7/2**	**0/6/3**	**0/7/2**	

Table 6 IGD values obtained for each algorithm on the test sets of each dataset

	MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
German	0.2028±	0.2191±	0.2201±	**0.1787**±	0.1880±	0.1852±	0.1906±	0.1903±
	0.0190↓	0.0296↓	0.0130↓	0.0263∘	**0.0101**∘	0.0159∘	0.0116∘	0.0131
Sonar	0.1866±	0.2034±	0.2774±	0.1535±	0.1762±	**0.1506**±	0.1687±	0.1548±
	0.0278↓	0.0301↓	0.0276↓	0.0326∘	0.0372↓	**0.0241**∘	0.0257↓	0.0269
Hill valley	0.0966±	0.1185±	0.2590±	0.0722±	0.0869±	0.0622±	0.0753±	**0.0486**±
	0.0212↓	0.0288↓	0.0298↓	0.0228↓	0.0272↓	**0.0160**↓	0.0212↓	0.0178
Musk1	0.0827±	0.1633±	0.2332±	0.0794±	0.0761±	0.0637±	0.0609±	**0.0544**±
	0.0183↓	0.0260↓	0.0187↓	0.0227↓	0.0357↓	**0.0099**↓	0.0150∘	0.0151
LSVT	0.2203±	0.3191±	0.3257±	0.1434±	0.2156 ±	**0.0891**±	0.1300±	0.1203±
	0.0610↓	0.0468↓	**0.0181**↓	0.0698↓	0.0674↓	0.0191↑	0.0649∘	0.0473
Madelon	0.1737±	0.3143±	0.4298±	0.0866±	0.1370±	0.2685±	0.1050±	**0.0961**±
	0.0257↓	0.0202↓	0.0126↓	0.0201↑	0.0522↓	0.0140↓	0.0160↓	**0.0098**
Isolet5	0.0814±	0.2309±	0.2830±	0.1043±	0.0817±	0.1461±	0.0835±	**0.0704**±
	0.0127↓	0.0181↓	0.0086↓	0.0201↓	0.0377∘	**0.0108**↓	0.0132↓	0.0111
Multiple	0.1219±	0.2463±	0.3106±	0.1260±	0.1071±	0.0985±	0.0877±	**0.0799**±
Features	0.0139↓	0.0131↓	0.0089↓	0.0151↓	0.0211↓	0.0106↓	0.0082↓	**0.0068**
CNAE	0.2089±	0.3583±	0.3618±	0.2154±	0.1423±	0.2189±	**0.1231**±	0.1333±
	0.0237↓	0.0085↓	0.0116↓	0.0231↓	0.0399∘	**0.0117**↓	0.0183↑	0.0302
↑,↓,∘	**0/9/0**	**0/9/0**	**0/9/0**	**0/7/2**	**0/6/3**	**0/7/2**	**1/5/3**	

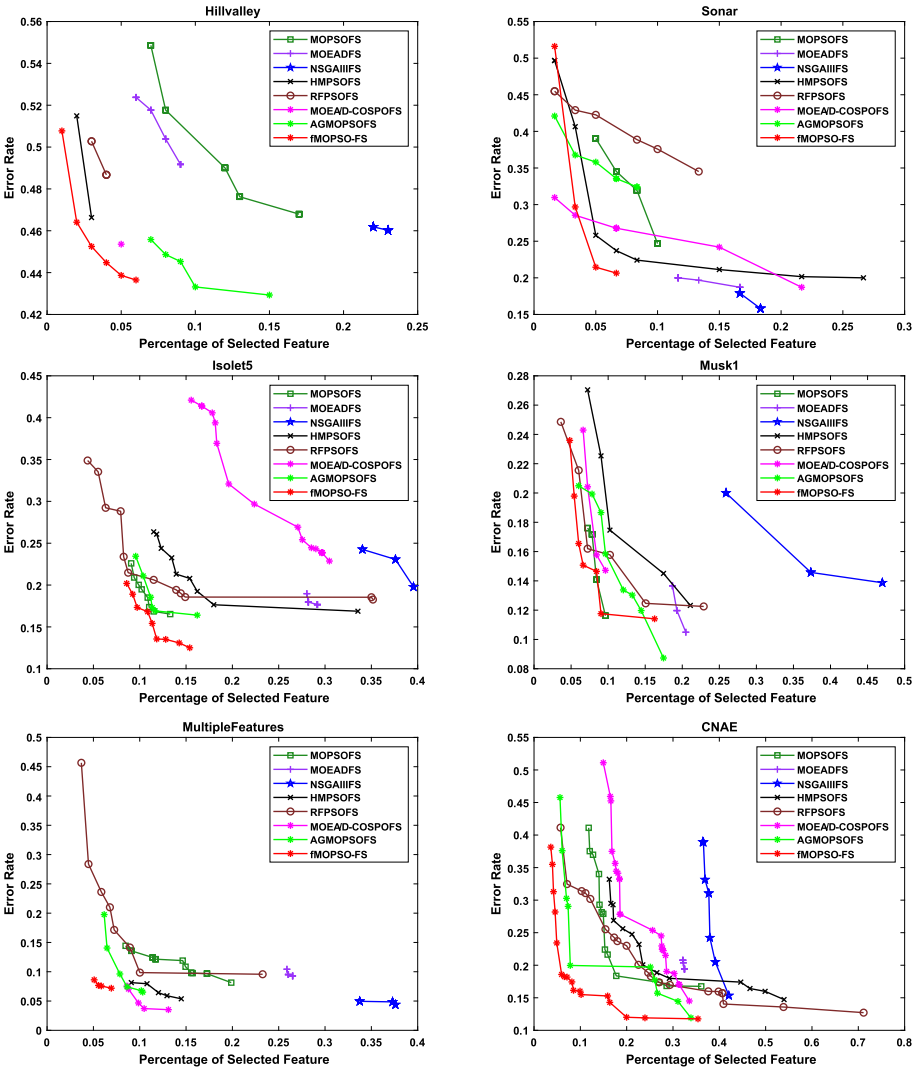

Fig. 2 Pareto fronts of the maximum HV value in the training set for each algorithm for each dataset

To visually verify the aforementioned conclusions, Figs. 2, and 3 show the PF with the maximum HV values obtained by each algorithm for 30 experiments on the training and test sets of each dataset, respectively. In Fig. 2, it is evident that fMOPSO-FS is able to obtain a set of Pareto solution sets with low classification error rates and compact feature subset sizes across the majority of the datasets. For the MultipleFeatures dataset, although the diversity of solutions obtained by the fMOPSO-FS algorithm is not as extensive as the other comparative algorithms, the fMOPSO-FS algorithm obtains candidate solutions of better quality than the other comparison algorithms. Moving to Fig. 3, compared to other algorithms on the test set, fMOPSO-FS continues to show its unique advantages. When the same number of features are selected, the fMOPSO-FS algorithm outperforms both HMPSOFS and MOEA/D-COPSO on the Isolet5 dataset. While HMPSOFS and MOEA/D-COPSO exhibit a broader

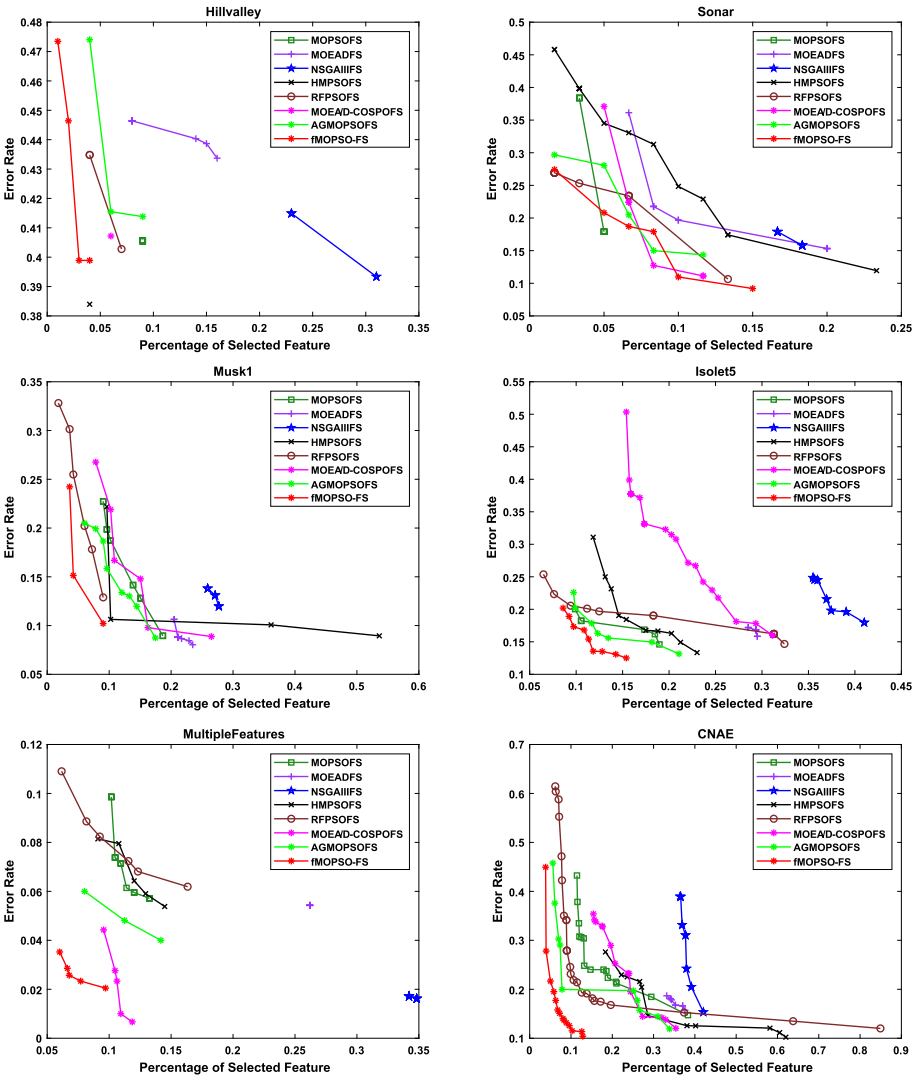

Fig. 3 Pareto fronts of the maximum HV value in the test set for each algorithm for each dataset

distribution of solution sets, their classification error rates are considerably higher compared to the fMOPSO-FS algorithm.

4.4 Comparison of Seven Feature Selection Methods using Different Classifiers

To verify the effectiveness of the proposed FS approach in this paper, this section classifies a subset of the features obtained by all the algorithms using the classical classifiers SVM, Naive Bayes and KNN, and the classification results are given in the Tables 7, 8, 9. In these tables, the mean classification precision on the training set is denoted by Tr_{Acc}, while Te_{Acc} represents the mean classification precision on the test set.

Table 7 The median value of the classification accuracy achieved by each algorithm on the SVM classifier

		MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COSPO	AGMOPSO	fMOPSO-FS
German	Tr_Acc	0.7131	0.7019	0.7186	0.7177	0.7159	0.7162	0.7234	0.7245
	Te_Acc	0.7087	0.7016	0.7137	0.7137	0.7102	0.7106	0.7174	0.7147
Sonar	Tr_Acc	0.7694	0.7856	0.8196	0.7415	0.7566	0.7780	0.7641	0.7850
	Te_Acc	0.7356	0.7416	0.7350	0.7057	0.7199	0.7383	0.7352	0.7363
Hillvalley	Tr_Acc	0.5516	0.5513	0.5767	0.5447	0.5464	0.5404	0.5539	0.5546
	Te_Acc	0.5114	0.5064	0.5360	0.5044	0.5051	0.5002	0.5135	0.5169
Musk1	Tr_Acc	0.8002	0.8420	0.8731	0.8178	0.7836	0.7989	0.7948	0.7859
	Te_Acc	0.7524	0.7771	0.7902	0.7665	0.7357	0.7502	0.7501	0.7449
LSVT	Tr_Acc	0.8907	0.9523	0.9941	0.8503	0.9005	0.9582	0.7752	0.6855
	Te_Acc	0.7726	0.7770	0.7962	0.7597	0.7891	0.8255	0.7432	0.6663
Madelon	Tr_Acc	0.6255	0.6723	0.6846	0.6267	0.6387	0.6454	0.6266	0.6244
	Te_Acc	0.5676	0.5610	0.5577	0.6010	0.5796	0.5761	0.6030	0.6080
Isolet5	Tr_Acc	0.9940	1	1	0.9988	0.9819	0.9911	1	1
	Te_Acc	0.8483	0.8974	0.8971	0.8686	0.8437	0.8199	0.9020	0.9031
Multiple	Tr_Acc	0.9999	1	1	0.9996	0.9994	1	1	1
Features	Te_Acc	0.9543	0.9684	0.9735	0.9586	0.9570	0.9617	0.9602	0.9654
CNAE	Tr_Acc	0.8286	0.9240	0.8631	0.9107	0.7546	0.8607	0.8714	0.8980
	Te_Acc	0.7659	0.8361	0.7512	0.8344	0.6946	0.7495	0.7850	0.8015

Table 8 The median value of the classification accuracy achieved by each algorithm on the Naive Bayes classifier

		MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
German	Tr_Acc	0.7197	0.7035	0.7085	0.7260	0.7245	0.7266	0.7281	0.7292
	Te_Acc	0.6977	0.6940	0.6898	0.7026	0.6992	0.6997	0.7003	0.7052
Sonar	Tr_Acc	0.6963	0.6930	0.6881	0.6798	0.6844	0.7057	0.7285	0.7235
	Te_Acc	0.7092	0.7038	0.6887	0.6881	0.7002	0.7015	0.7256	0.7118
Hillvalley	Tr_Acc	0.5217	0.5224	0.5232	0.5195	0.5208	0.5210	0.5202	0.5195
	Te_Acc	0.4804	0.4751	0.4732	0.4849	0.4773	0.4780	0.4830	0.4824
Musk1	Tr_Acc	0.6786	0.6972	0.7092	0.6871	0.6678	0.6826	0.6879	0.6715
	Te_Acc	0.5891	0.5887	0.6120	0.5832	0.5744	0.5756	0.5700	0.5690
LSVT	Tr_Acc	0.4946	0.4981	0.5031	0.4823	0.4883	0.5469	0.4739	0.5619
	Te_Acc	0.5581	0.5503	0.5509	0.5354	0.5500	0.5727	0.5286	0.5422
Madelon	Tr_Acc	0.6475	0.6684	0.6929	0.6092	0.6357	0.6429	0.6097	0.6035
	Te_Acc	0.6109	0.6226	0.6031	0.6180	0.6152	0.6086	0.6235	0.6263
Isolet5	Tr_Acc	0.8447	0.8843	0.8838	0.8612	0.8459	0.8495	0.8867	0.8825
	Te_Acc	0.7736	0.8081	0.8096	0.7861	0.7740	0.7586	0.7818	0.7823
Multiple	Tr_Acc	0.9584	0.9729	0.9708	0.9646	0.9602	0.9690	0.9771	0.9821
Features	Te_Acc	0.9019	0.9109	0.9151	0.9034	0.9048	0.9322	0.9061	0.9157
CNAE	Tr_Acc	0.7448	0.8409	0.7816	0.7968	0.6881	0.7305	0.8113	0.8245
	Te_Acc	0.6839	0.7602	0.6936	0.7553	0.6291	0.6737	0.7478	0.7596

Table 9 The median value of the classification accuracy achieved by each algorithm on the KNN classifier

		MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
German	Tr_Acc	0.7013	0.7310	0.7386	0.7245	0.7338	0.6367	0.7186	0.7211
	Te_Acc	0.6859	0.7170	0.7192	0.7057	0.7112	0.6191	0.6971	0.7096
Sonar	Tr_Acc	0.7363	0.8416	0.8123	0.6798	0.6844	0.7859	0.8143	0.8093
	Te_Acc	0.7187	0.8253	0.7890	0.7447	0.7358	0.7657	0.7940	0.7870
Hillvalley	Tr_Acc	0.5437	0.5528	0.5230	0.5491	0.6032	0.5416	0.5610	0.5715
	Te_Acc	0.5238	0.5376	0.5001	0.5205	0.5850	0.5220	0.5457	0.5554
Musk1	Tr_Acc	0.8486	0.8762	0.8977	0.8643	0.8412	0.8432	0.8876	0.8970
	Te_Acc	0.8069	0.8270	0.8447	0.8379	0.7867	0.8026	0.8417	0.8568
LSVT	Tr_Acc	0.5547	0.6174	0.7625	0.7932	0.6662	0.8235	0.6672	0.7109
	Te_Acc	0.6232	0.7012	0.8264	0.8072	0.7430	0.8903	0.7725	0.8785
Madelon	Tr_Acc	0.7863	0.7784	0.6621	0.8322	0.7349	0.6791	0.7858	0.8340
	Te_Acc	0.6994	0.7618	0.6583	0.8430	0.7357	0.5516	0.7715	0.8457
Isolet5	Tr_Acc	0.8537	0.8427	0.8335	0.8702	0.8152	0.8058	0.8706	0.8789
	Te_Acc	0.8074	0.8210	0.7626	0.8438	0.7625	0.7252	0.8185	0.8256
Multiple	Tr_Acc	0.9482	0.9034	0.9883	0.9045	0.9512	0.9396	0.9670	0.9721
Features	Te_Acc	0.9113	0.8809	0.9639	0.8778	0.9298	0.9112	0.9321	0.9369
CNAE	Tr_Acc	0.7453	0.8613	0.7823	0.8412	0.7483	0.7305	0.7528	0.7634
	Te_Acc	0.7013	0.7310	0.7386	0.7245	0.7338	0.6367	0.7186	0.7211

An observation from Tables 7 and 8 reveals that for the datasets German, Ioslet5 and MultipleFeatures, the feature subset selected by fMOPSO-FS shows better classification performance on SVM and Naive Bayes classifiers than other comparison algorithms. Although MOEA/D and NSGAIII can demonstrate superior classification accuracy in most datasets, it becomes apparent from Figs. 2 and 3 that it is difficult to find a set of solutions with good diversity using these two algorithms, and the feature subset selected by these algorithm are significantly larger in size compared to the feature subset chosen by fMOPSO-FS. In Table 7, the classification accuracy of the fMOPSO-FS algorithm on the LSVT dataset differs significantly from other comparison algorithms. This may be due to the low correlation between features and categories in the LSVT dataset, resulting in the selection of representative features and poor classification performance of the algorithm. According to Table 9, the feature subset obtained by MOEA/D demonstrates superior classification performance on most of the datasets. However, for the Hillvalley dataset, the feature subset chosen by RFPSOFS performs better than other comparison algorithms, while for the Musk1 and Isolet5 datasets, fMOPSO-FS exhibits better classification performance. Table 10 shows the average number of features selected in the subset of features selected by each algorithm. From Table 10, it can be seen that on most datasets, the fMOPSO algorithm selects fewer features than other comparison algorithms, except on the German dataset. Although some comparison algorithms have achieved better classification accuracy on some datasets, taking into account the number of selected features, diversity of candidate solutions and the evaluation results of HV and IGD indicators, the fMOPSO-FS algorithm still has better performance compared with other algorithms.

4.5 Experimental Analysis on Gene Expression Datasets

The preceding subsection showcases the satisfactory performance of the proposed algorithm on conventional datasets, which are typically characterized by low feature dimensionality and a large number of samples. To verify that the fMOPSO-FS algorithm can also demonstrate its advantages on high-dimensional datasets, therefore, we selected six gene expression profile datasets, Colon, SRBCT, Lymphoma, Leukemia3, Lung and Kolod, which have high latitude and a small number of instances. Table 11 [40, 41] presents the specifications and details of the datasets used in the paper. For testing, Tables 12 and 13 show the HV values obtained by the fMOPSO-FS algorithm and other comparison algorithms on the training and test sets of the above six datasets. From the perspective of the training set, in Table 12, the fMOPSO-FS algorithm obtained HV values on the Colon and Lung datasets that were close to those of the AGMOPSO algorithm, and achieved better HV values compared to other comparative algorithms. On the SRCBT and Leukemia3 datasets, the HV values obtained by fMOPSO-FS were only slightly lower than those obtained by the AGMOPSO algorithm. The HV value obtained by fMOPSO-FS on the Lung dataset is only slightly lower than that obtained by the MOEA/D-COPSO algorithm. On the Kolod dataset, fMOPSO-FS achieved better HV values than other comparison algorithms. From the perspective of the test set, in Table 13, the HV values obtained by fMOPSO-FS and AGMOPSO algorithm are similar on most datasets, but lower than those obtained by MOEA/D-COPSO algorithm on the lung dataset. The HV values obtained from the Leukemia3 dataset are close to those obtained from HMPSOFS and RFPSOFS algorithms. In summary, it can be inferred that the fMOPSO-FS algorithm can also perform well on high-dimensional datasets.

Table 10 Average number of selected features for each algorithm on different datasets

	MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
German	3.65	**2.12**	6.56	4.05	3.43	3.23	4.68	4.38
Sonar	7.99	9.80	18.50	6.14	7.29	7.80	6.12	**5.29**
Hill valley	13.16	14.25	32.20	9.86	11.88	9.71	11.53	**6.81**
Musk1	26.13	40.04	57.16	33.25	25.38	27.09	22.63	**21.42**
LSVT	39.49	77.44	111.44	24.06	40.00	38.53	11.66	**11.29**
Madelon	81.01	156.56	183.42	36.42	79.78	91.53	38.05	**26.21**
Isolet5	97.81	193.60	231.05	131.99	114.09	146.60	101.64	**88.70**
Multiple Features	102.54	192.10	240.36	123.53	113.57	102.71	77.83	**62.97**
CNAE	186.09	303.02	328.32	292.46	183.91	203.14	238.54	**104.51**

Table 11 Details of relevant gene expression profile datasets

Dataset	Number of features	Number of records	Number of classes
Colon	2000	62	2
SRBCT	2308	83	4
Lymphoma	4026	66	2
Leukemia3	7129	72	2
Lung	7129	96	2
Kolod	10686	704	3

4.6 Parameter Analysis

The proposed initialization strategy contains two adjustment factors α and β. As the selection threshold is fixed at 0.6 and β affects the upper bound of the coding interval for less relevant features, β taken at 0.4 enables the feature coding interval to range from [0, 1] in line with the random initialization interval, while also ensuring that features with higher relevance have a higher selection probability. Thus, this section focuses on the effect of α on particle mass and particle distribution. The Fig. 4 depicts the initialization process of the fMOPSO-FS algorithm on various datasets with different values of α. The graph reveals a gradual decline in the quality of the generated initial solutions as α increases. Hence, it is unnecessary to set a large α value, while a small α value has a negligible impact on the population. Consequently, we define the range of α selection as 0.1, 0.2, 0.3, 0.4, 0.5. Notably, when α is set to 0.1, the particles in the target space are positioned closer to the origin, which means that the obtained initial solutions have higher quality. Therefore, the value of α is set to 0.1.

4.7 Analysis of the Proposed Strategies

To further analyze the effectiveness of the algorithm, the proposed feature-label correlation-guided initialization strategy and adaptive perturbation strategy as well as the introduced mutual information theory are validated separately.

4.7.1 Initialization Strategy Analysis

In order to assess the effectiveness of the initialization strategy, we compared it with the random initialization strategy. In Fig. 5, PF represents the Pareto front simulated by the non-dominated solutions produced by fMOPSO-FS and all the comparison algorithms after 30 times independent runs. It can be clearly seen from Fig. 5 that the initial solution generated by the proposed initialization strategy on most data sets exhibits greater proximity to the Pareto front. Although the impact on the Musk1 and SRBCT datasets is not significant, it is likely due to the limited correlation between the feature data and class labels. However, it is worth noting that this hybrid initialization method still effectively enhances the diversity of initial solutions compared to a single initialization method. The initial solutions generated by a single initialization method will be limited to a certain area in the target space, while using a mixed initialization method will cover more areas and expand the search range of particles.

Table 12 HV values obtained for each algorithm on the training sets of each dataset

	MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
Colon	0.8018±	0.5994±	0.5729±	0.8328±	0.7909±	0.7957±	0.8616±	**0.8690**±
	0.0301↓	0.0281↓	**0.0209**↓	0.0306↓	0.0659↓	0.0355↓	0.0271∘	0.0300
SRBCT	0.8824±	0.6466±	0.5499±	0.9039±	0.8753±	0.8265±	**0.9449**±	0.9343±
	0.0245↓	0.0183↓	0.0283↓	0.0196↓	0.0441↓	0.0221↓	**0.0083**↑	0.0145
Lymphoma	0.8638±	0.6886±	0.6443±	0.9478±	0.9478±	0.9487±	0.9630±	**0.9726**±
	0.0257↓	0.0133↓	0.0100↓	0.0114↓	0.0198↓	0.0306↓	**0.0098**↓	0.0192
Leukemia3	0.8285±	0.6255±	0.5110±	0.8720±	0.8685±	0.7776±	**0.9308**±	0.8505±
	0.0244↓	0.0198↓	0.0192↓	0.0197↑	0.0538↑	0.0259↓	**0.0155**↑	0.0234
Lung	0.9009±	0.6651±	0.6468±	0.9337±	0.9040±	**0.9849**±	0.9633±	0.9643±
	0.0116↓	0.0079↓	0.0037↓	0.0189↓	0.0282↓	0.0111↑	**0.0069**∘	0.0096
Kolod	0.8742±	0.6340±	0.6296±	0.8635±	0.9088±	0.8473±	0.9170±	**0.9247**±
	0.0180↓	0.0092↓	0.0087↓	0.0233↓	0.0338↓	0.0188↓	0.0148↓	**0.0077**
↑,↓,∘	**0/6/0**	**0/6/0**	**0/6/0**	**0/6/0**	**0/6/0**	**0/6/0**	**2/2/2**	

Table 13 HV values obtained for each algorithm on the test sets of each dataset

	MOPSO	MOEA/D	NSGAIII	HMPSOFS	RFPSOFS	MOEA/D-COPSO	AGMOPSO	fMOPSO-FS
Colon	0.7352±	0.5650±	0.5147±	0.7641±	0.7461±	0.7323±	0.7972±	**0.8127**±
	0.0760↓	**0.0418**↓	0.0598↓	0.0638↓	0.0738↓	0.0579↓	0.0665∘	0.0728
SRBCT	0.8240±	0.6139±	0.5528±	0.8663±	0.8361±	0.7717±	**0.9018**±	0.8908±
	0.0580↓	0.0439↓	0.0595↓	0.0396↓	0.0650↓	0.0616↓	**0.0264**∘	0.0472
Lymphoma	0.8555±	0.6843±	0.6366±	0.9015±	0.9113±	0.9142±	0.9428±	**0.9484**±
	0.0284↓	**0.0172**↓	0.0190↓	0.0256↓	0.0301↓	0.0372↓	0.0245∘	0.0332
Leukemia3	0.7554±	0.5766±	0.4916±	0.7909±	0.8275±	0.7048±	**0.8585**±	0.7944±
	0.0625↓	0.0534↓	0.0531↓	0.0555∘	0.0761∘	0.0819↓	**0.0493**↑	0.0794
Lung	0.8797±	0.6555±	0.6437±	0.9161±	0.8902±	**0.9744**±	0.9415±	0.9393±
	0.0220↓	0.0114↓	**0.0084**↓	0.0275↓	0.0318↓	0.0169↑	0.0199∘	0.0258
Kolod	0.8589±	0.6335±	0.6224±	0.8542±	0.8987±	0.8352±	0.9085±	**0.9159**±
	0.0213↓	0.0121↓	0.0111↓	0.0227↓	0.0340↓	0.0189↓	0.0201↓	**0.0075**
↑,↓,∘	**0/6/0**	**0/6/0**	**0/6/0**	**0/5/1**	**0/5/1**	**1/5/0**	**1/1/4**	

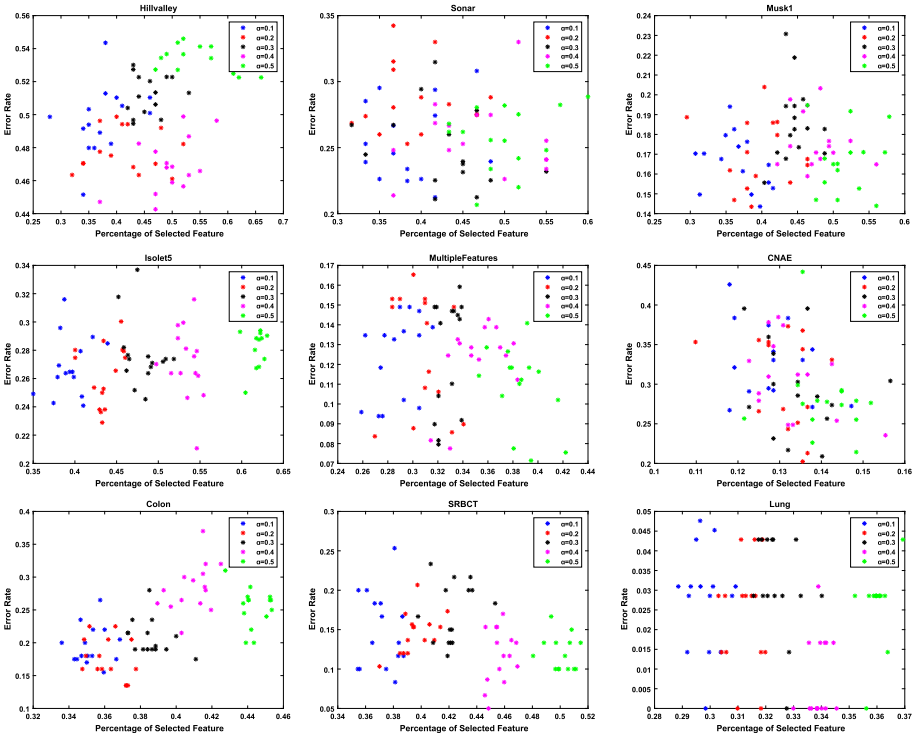

Fig. 4 Distribution of particle populations on different datasets with different α values

4.7.2 Analysis of Adaptive Hybrid Perturbation Strategies

To validate the efficacy of the adaptive hybrid perturbation strategy, the MOPSO-FL-FS algorithm retained only the feature-label-guided initialization strategy, and the fMOPSO-FS-F algorithm retained the feature-label-guided initialization strategy while using a perturbation strategy with a fixed probability of variation. Each algorithm is run independently for 30 times, and the Tables 14 and 15 show the HV values of the algorithms on the training and test sets of each dataset, respectively. In the Tables 14 and 15, it can be seen that the HV values obtained by the fMOPSO-FS algorithm are significantly better than those obtained by the MOPSO-FL-FS algorithm on both the training and test sets, and compared with the fMOPSO-FS-F algorithm, although the HV values obtained on most of the datasets are the HV values obtained are similar, the advantage is also demonstrated on some of the datasets. One possible reason for this is that the adaptive hybrid perturbation strategy can dynamically adjust the variation probability based on the performance of the particles themselves, unlike fixed variation probability. This adaptiveness allows the strategy to fine-tune and optimize the exploration and exploitation trade-off, leading to improved results.

4.7.3 Validation of the Validity of the Mutual Information Theory

The validity of the MI theory is also validated for the different datasets. In this section, the subset of features obtained by running the fMOPSO-FS algorithm 30 times independently on each of these datasets are analysed, and the Fig. 6 shows how well the algorithm works

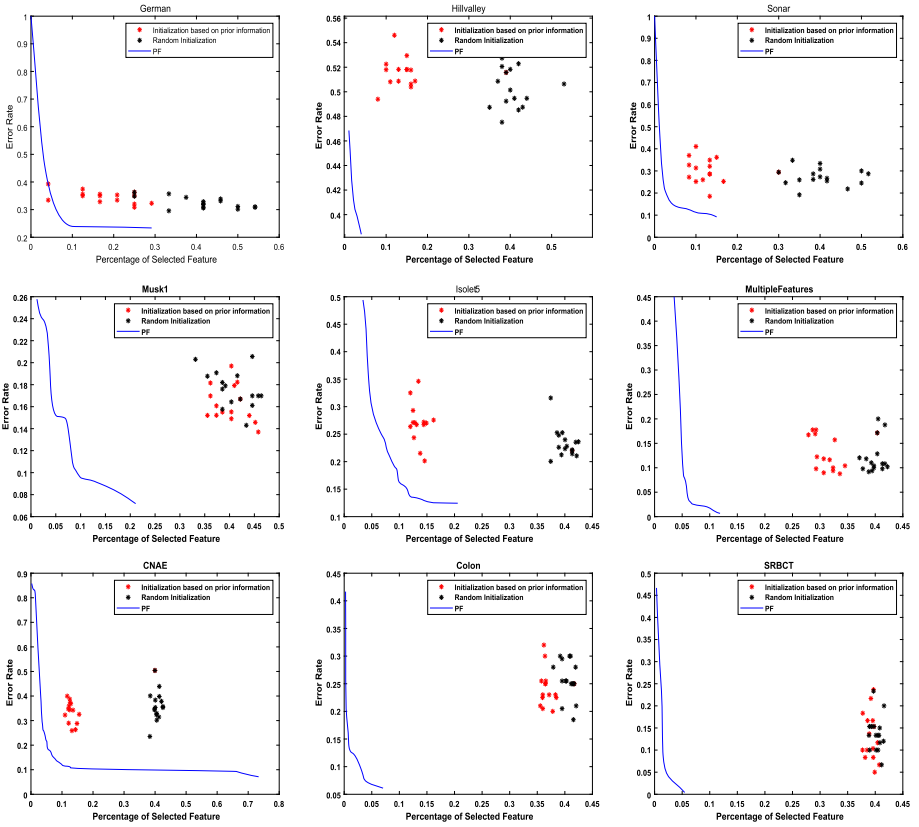

Fig. 5 Distribution of particles with different initialization strategies on different datasets

Table 14 HV values obtained for each algorithm on the training sets of each dataset

	MOPSO-FL-FS	fMOPSO-FS-F	fMOPSO-FS
German	0.7300±0.0107↓	0.7416±0.0102↓	**0.7465**±0.0075
Sonar	0.7965±0.0295↓	**0.8768**±0.0133∘	0.8729±0.0172
Hill valley	0.5531±0.0280↓	**0.6264**±0.0195∘	0.6239±0.0144
Musk1	0.7905±0.0196↓	0.8571±0.0135↓	**0.8660**±0.0186
LSVT	0.7677±0.0137↓	0.8328±0.0142∘	**0.8339**±0.0190
Madelon	0.7895±0.0167↓	0.8024±0.0203↓	**0.8365**±0.0141
Isolet5	0.7175±0.0228↓	0.7771±0.0150↓	**0.7914**±0.0138
MultipleFeatures	0.8504±0.0153↓	**0.9060**±0.0088∘	0.9021±0.0096
CNAE	0.7926±0.0155↓	**0.8602**±0.0110∘	0.8562±0.0081
↑,↓,∘	**0/9/0**	**0/4/5**	

Table 15 HV values obtained for each algorithm on the test sets of each dataset

	MOPSO-FL-FS	fMOPSO-FS-F	fMOPSO-FS
German	0.7099±0.0168∘	**0.7204**±0.0152 ∘	0.7143±0.0171
Sonar	0.7397±0.0444↓	0.7936±0.0321∘	**0.8078**±0.0296
Hill valley	0.5266±0.0274↓	0.5698±0.0330↓	**0.5844**±0.0207
Musk1	0.7564±0.0318↓	0.8189±0.0202∘	**0.8252**±0.0272
LSVT	0.7328±0.0189↓	0.7536±0.0216∘	**0.7464**±0.0349
Madelon	0.7573±0.0178↓	0.7824±0.0184↓	**0.8277**±0.0205
Isolet5	0.7018±0.0254↓	0.7599±0.0139↓	**0.7745**±0.0168
MultipleFeatures	0.8395±0.0209↓	**0.8872**±0.0134∘	0.8834±0.0168
CNAE	0.7824±0.0244↓	**0.8445**±0.0173∘	0.8387±0.0172
↑,↓,∘	**0/8/1**	**0/3/6**	

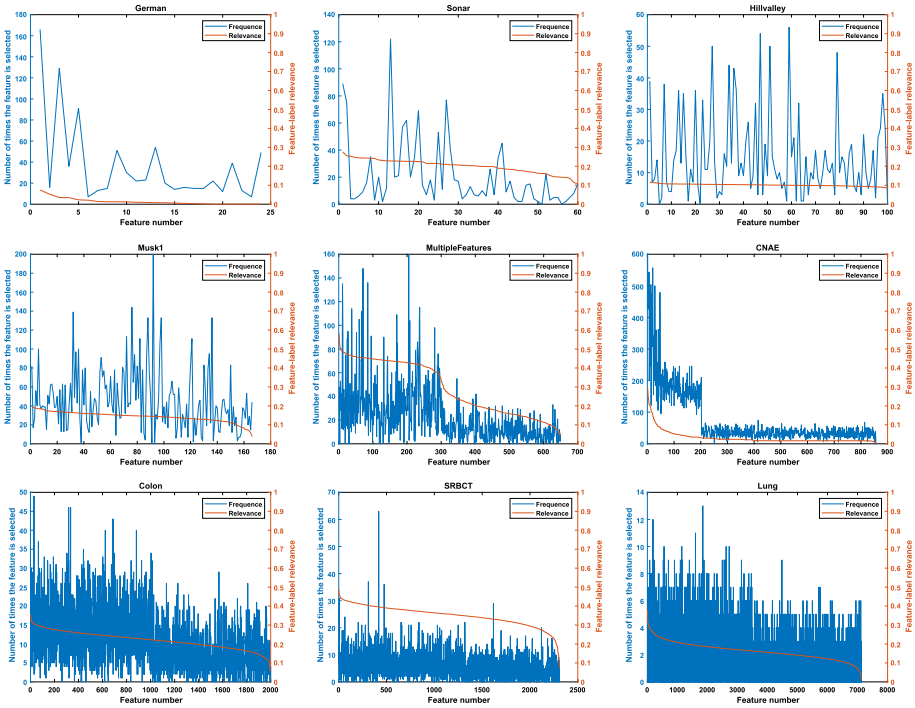

Fig. 6 Statistics on the frequency of feature selection

on these datasets in terms of the number of times each feature was selected and the statistics of how often they correlate with the categories.

In Fig. 6 the X-axis denotes the feature ordinal number, determined by ranking their relevance to class labels from highest to lowest. The left and right Y-axis represent the frequency of FS and the correlation between features and categories, respectively. From Fig. 6, it is evident that on the MultipleFeature and CNAE datasets, as the correlation between a feature and the label increases, the frequency of selecting that particular feature also increases.

Conversely, as the correlation decreases, the corresponding features are less selected. On the gene expression profile datasets Colon, SRBCT and Lung, the correlation and feature frequency trends of the selected features are roughly the same. For other datasets, since the correlation of features and labels is not so obvious, there is no obvious regularity in the presented images. It can be concluded that the incorporation of prior information can guide the algorithm to select features with higher relevance to the class labels, thus obtaining a higher quality subset of features, and can enhance the explainability of the selected features.

5 Conclusions

The randomness and lack of knowledge guiding the initialization process of most existing MOPSO-based feature selection methods may lead to the initialized solutions searching or even repeating meaningless regions of the search space during the evolution process, and the generated initial population may be far from the true Pareto front. Furthermore, the absence of sufficient selection pressure on the particle population during the later stages of iterative evolution results in a predisposition for the population to converge towards local optima. In order to enhance the distribution of the initial population and the quality of the initial solutions, while avoiding the particle swarm from being stuck in local optima, an adaptive MOPSOFS method based on feature-label correlation guidance is proposed in this paper. The method adopts a novel initialization strategy that makes full use of the prior knowledge in the feature data to obtain higher quality initial solutions. Simultaneously, an adaptive hybrid mutation strategy is proposed to enable the particle swarm to escape local optima. This mutation strategy dynamically adjusts the mutation rate based on the convergence status of the swarm, facilitating exploration of the search space and reducing the likelihood of getting trapped in suboptimal solutions.

The experimental findings validate that the method has greater advantages in solving the multi-objective FS problem, but there are still some problems to be tackled. Firstly, it can be time-consuming in obtaining prior information on feature data, especially for datasets with large feature dimensions. Secondly, the method mainly considers the correlation between features and categories, so there may still be a small number of redundant features in the obtained feature subset. Hence, improving the efficiency of obtaining certain prior information, combining the correlation between features and further eliminating redundant features should be the primary areas of focus for future research.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant Nos. 61976108 and 61572241.

Data availability and access All data generated or analysed during this study are available from the corresponding author on reasonable request.

Declarations

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

Ethical and informed consent for data used This article does not contain any studies with human participants or animals performed, and the datasets used are publicly available online.

References

1. Jiao R, Xue B, Zhang M (2022) Benefiting from single-objective feature selection to multiobjective feature selection: a multiform approach. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2022.3218345
2. Zebari R, Abdulazeez A, Zeebaree D, Zebari D, Saeed J (2020) A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. J Appl Sci Technol Trends 1(2):56–70. https://doi.org/10.38094/jastt1224
3. Liang JJ, Baskar S, Suganthan PN, Qin AK (2006) Performance evaluation of multiagent genetic algorithm. Nat Comput 5:83–96. https://doi.org/10.1007/s11047-005-1625-y
4. Qiu H, Xia X, Li Y, Deng X (2023) A dynamic multipopulation genetic algorithm for multiobjective workflow scheduling based on the longest common sequence. Swarm Evol Comput 78:101291. https://doi.org/10.1016/j.swevo.2023.101291
5. Lin Q, Liu S, Zhu Q, Tang C, Song R, Chen J, Zhang J (2016) Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. IEEE Trans Evol Comput 22(1):32–46. https://doi.org/10.1109/TEVC.2016.2631279
6. Niu P, Niu S, Chang L (2019) The defect of the Grey Wolf optimization algorithm and its verification method. Knowl-Based Syst 171:37–43. https://doi.org/10.1016/j.knosys.2019.01.018
7. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4, pp 1942–1948. IEEE. https://doi.org/10.1109/ICNN.1995.488968
8. Wang F, Wang X, Sun S (2022) A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization. Inf Sci 602:298–312. https://doi.org/10.1016/j.ins.2022.04.053
9. Xiang Z, Shao X, Wu H, Ji D, Yu F, Li Y (2020) An adaptive integral separated proportional-integral controller based strategy for particle swarm optimization. Knowl-Based Syst 195:105696. https://doi.org/10.1016/j.knosys.2020.105696
10. Xia X, Song H, Zhang Y, Gui L, Xu X, Li K, Li Y (2022) A particle swarm optimization with adaptive learning weights tuned by a multiple-input multiple-output fuzzy logic controller. IEEE Trans Fuzzy Syst. https://doi.org/10.1109/TFUZZ.2022.3227464
11. Dhal P, Azad C (2021) A multi-objective feature selection method using newton's law based pso with gwo. Appl Soft Comput 107:107394. https://doi.org/10.1016/j.asoc.2021.107394
12. Wang Z, Li M, Li J (2015) A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure. Inf Sci 307:73–88. https://doi.org/10.1016/j.ins.2015.02.031
13. Xue Y, Zhu H, Neri F (2022) A self-adaptive multi-objective feature selection approach for classification problems. Integr Comput-Aided Eng 29(1):3–21. https://doi.org/10.3233/ICA-210664
14. Feng J, Gong Z (2022) A novel feature selection method with neighborhood rough set and improved particle swarm optimization. IEEE Access 10:33301–33312. https://doi.org/10.1109/ACCESS.2022.3162074
15. Han F, Wang T, Ling Q (2023) An improved feature selection method based on angle-guided multi-objective PSO and feature-label mutual information. Appl Intell 53(3):3545–3562. https://doi.org/10.1007/s10489-022-03465-9
16. Wang S, Yu X, Jia W (2021) A new population initialization of particle swarm optimization method based on pca for feature selection. J Big Data 3(1):1. https://doi.org/10.32604/jbd.2021.010364
17. Li X, Fu Q, Li Q, Ding W, Lin F, Zheng Z (2023) Multi-objective binary grey wolf optimization for feature selection based on guided mutation strategy. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2023.110558
18. Zhou S, Sha L, Zhu S, Wang L (2022) Adaptive hierarchical update particle swarm optimization algorithm with a multi-choice comprehensive learning strategy. Appl Intell. https://doi.org/10.1007/s10489-021-02413-3
19. Wei B, Wang X, Xia X, Jiang M, Ding Z, Huang Y (2021) Novel self-adjusted particle swarm optimization algorithm for feature selection. Computing. https://doi.org/10.1007/s00607-020-00891-w
20. Xiang Z, Ji D, Zhang H, Wu H, Li Y (2019) A simple PID-based strategy for particle swarm optimization algorithm. Inf Sci 502:558–574. https://doi.org/10.1016/j.ins.2019.06.042

21. Li S, Wang F, He Q, Wang X (2023) Deep reinforcement learning for multi-objective combinatorial optimization: a case study on multi-objective traveling salesman problem. Swarm Evol Comput 83:101398. https://doi.org/10.1016/j.swevo.2023.101398
22. Rashno A, Shafipour M, Fadaei S (2022) Particle ranking: an efficient method for multi-objective particle swarm optimization feature selection. Knowl-Based Syst 245:108640. https://doi.org/10.1016/j.knosys.2022.108640
23. Jiang J, Han F, Ling Q, Wang J, Li T, Han H (2020) Efficient network architecture search via multiobjective particle swarm optimization based on decomposition. Neural Netw 123:305–316. https://doi.org/10.1016/j.neunet.2019.12.005
24. Liu P, Liu J (2017) Multi-leader PSO (MLPSO): a new PSO variant for solving global optimization problems. Appl Soft Comput 61:256–263. https://doi.org/10.1016/j.asoc.2017.08.022
25. Bonnlander BV, Weigend AS (1994) Selecting input variables using mutual information and nonparametric density estimation. In: Proceedings of the 1994 international symposium on artificial neural networks (ISANN'94), pp 42–50. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0cbb68b53c3a079545790e1e97e9f14bb4d613a9
26. Han F, Chen WT, Ling QH, Han H (2021) Multi-objective particle swarm optimization with adaptive strategies for feature selection. Swarm Evol Comput 62:100847. https://doi.org/10.1016/j.swevo.2021.100847
27. Zhang Y, Gong DW, Cheng J (2015) Multi-objective particle swarm optimization approach for cost-based feature selection in classification. IEEE/ACM Trans Comput Biol Bioinf 14(1):64–75. https://doi.org/10.1109/TCBB.2015.2476796
28. Zapotecas Martínez S, Coello Coello CA (2011) A multi-objective particle swarm optimizer based on decomposition. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation, pp 69–76 https://doi.org/10.1145/2001576.2001587
29. Chauhan S, Singh M, Aggarwal AK (2023) Investigative analysis of different mutation on diversity-driven multi-parent evolutionary algorithm and its application in area coverage optimization of WSN. Soft Comput. https://doi.org/10.1007/s00500-023-08090-3
30. Khalfi S, Iacca G, Draa A (2022) On the use of single non-uniform mutation in lightweight metaheuristics. Soft Comput 26(5):2259–2275. https://doi.org/10.1007/s00500-021-06495-6
31. Zhao J, Chen DD, Xiao R (2022) A heterogeneous variation firefly algorithm with maximin strategy. CAAI Trans Intell Syst 17(1):116–130. https://doi.org/10.11992/tis.202106018 https://doi.org/10.11992/tis.202106018
32. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279. https://doi.org/10.1109/TEVC.2004.826067
33. Deb K, Jain H (2013) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601. https://doi.org/10.1109/TEVC.2013.2281535
34. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731. https://doi.org/10.1109/TEVC.2007.892759
35. Amoozegar M, Minaei-Bidgoli B (2018) Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism. Expert Syst Appl 113:499–514. https://doi.org/10.1016/j.eswa.2018.07.013
36. Zhou Y, Kang J, Guo H (2020) Many-objective optimization of feature selection based on two-level particle cooperation. Inf Sci 532:91–109. https://doi.org/10.1016/j.ins.2020.05.004
37. Shang K, Ishibuchi H, He L, Pang LM (2020) A survey on the hypervolume indicator in evolutionary multiobjective optimization. IEEE Trans Evol Comput 25(1):1–20. https://doi.org/10.1109/TEVC.2020.3013290
38. Wu B, Hu W, Hu J, Yen GG (2019) Adaptive multiobjective particle swarm optimization based on evolutionary state estimation. IEEE Trans Cybern 51(7):3738–3751. https://doi.org/10.1109/TCYB.2019.2949204
39. Dua D Karra Taniskidou E (2017) UCI Machine Learning Repository. Irvine: University of California, School of Information and Computer Science. https://archive.ics.uci.edu/ml
40. Zhu Z, Ong YS, Dash M (2007) Markov blanket-embedded genetic algorithm for gene selection. Pattern Recogn 40(11):3236–3248. https://doi.org/10.1016/j.patcog.2007.02.007
41. Kolodziejczyk AA, Kim JK, Tsang JC, Ilicic T, Henriksson J, Natarajan KN, Teichmann SA (2015) Single cell RNA-sequencing of pluripotent states unlocks modular transcriptional variation. Cell Stem Cell 17(4):471–485. https://doi.org/10.1016/j.stem.2015.09.011