Swarm and Evolutionary Computation 91 (2024) 101743

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A leader-adaptive particle swarm optimization with dimensionality reduction strategy for feature selection

Shanshan Yang [a,c], Bo Wei [a,c,*], Li Deng [b,c], Xiao Jin [a,c], Mingfeng Jiang [a], Yanrong Huang [d], Feng Wang [e]

[a] School of Computer Science and Technology (School of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou, 310018, China
[b] School of Science, Zhejiang Sci-Tech University, Hangzhou, 310018, China
[c] Longgang Research Institute, Zhejiang Sci-Tech University, Longgang, 325000, China
[d] College of Economics and Management, Zhejiang University of Water Resources and Electric Power, Hangzhou, 310018, China
[e] School of Computer Science, Wuhan University, Wuhan, 430072, China

ARTICLE INFO

Keywords:
Feature selection
Evolutionary computation
Particle swarm optimization
Leader-adaptive strategy
Markov blanket

ABSTRACT

Feature selection (FS) is a key data pre-processing method in machine learning tasks. It aims to obtain better classification accuracy of an algorithm with the smallest size of selected feature subset. Particle Swarm Optimization has been widely applied in FS tasks. However, when solving FS task on high-dimensional datasets, most of the PSO-based FS methods are easy to get premature convergence and fall into the local optimum. To address this issue, a leader-adaptive particle swarm optimization with dimensionality reduction strategy (LAPSO-DR) is proposed in this paper. Firstly, a hybrid initialization strategy based on feature importance is formulated. The population is divided into two parts, which have different initialization ranges. It can not only improve the diversity of the population but also eliminate some redundant features. Secondly, the leader-adaptive strategy is proposed to improve the exploitation ability of the population, in which each particle can have a different learning exemplar selected from the elite sub-swarm. Finally, the dimensionality reduction strategy based on Markov blanket is introduced to reduce the size of the optimal feature subset. LAPSO-DR is compared with 8 representative FS methods on 18 benchmark datasets. The experimental results show that LAPSO-DR can obtain smaller sizes of feature subsets with highest classification accuracies on 17 out of 18 datasets. The classification accuracies of LAPSO-DR are over 90% on 14 datasets and the feature elimination rates are higher than 60% on 18 datasets.

1. Introduction

In the era of big data, more and more high-dimensional datasets are emerging from various fields [1,2]. However, these high-dimensional datasets not only increase the computational complexity of an algorithm, but also negatively affect its performance. This is called the "curse of dimensionality" [3]. As an important data preprocessing technology, feature selection (FS) can reduce the dimensionality of these datasets by removing redundant and irrelevant features, with the purpose of improving the performance of learning algorithms and saving computation cost.

Currently, there are three categories of FS algorithms, namely, filter [4], wrapper, and embedded algorithms [5]. In general, filter algorithms evaluate the feature importance only by employing statistical methods, while wrapper algorithms evaluate the quality of a feature subset using a learning algorithm. Embedded algorithms consider the FS as part of the model training, which can be conducted automatically during the training process, completing the FS task during the learning process. In terms of time, the filter algorithm is the least time-consuming among the three methods when conducting the same tasks. However, the other two algorithms have significantly better performance than the filter one in terms of classification accuracy. Among them, wrapper algorithm has the best classification accuracy, so we use it in our work. But what cannot be ignored is that the wrapper algorithm may take a long time to find the optimal feature subset on the high-dimensional dataset. In fact, if the dimension of the dataset is

* Corresponding author at: School of Computer Science and Technology (School of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou, 310018, China.

E-mail address: weibo@zstu.edu.cn (B. Wei).

https://doi.org/10.1016/j.swevo.2024.101743
Received 27 February 2024; Received in revised form 16 July 2024; Accepted 17 September 2024
Available online 27 September 2024
2210-6502/© 2024 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

n, the number of evaluations increases exponentially to $2^n - 1$. Thus, performing FS tasks on high-dimensional datasets remains challenging.

In recent years, Evolutionary Algorithms (EA) have been successfully applied to various practical optimization problems [6–9]. Especially, they have been widely applied into FS tasks, which include Genetic Algorithm (GA) [10], Ant Colony Algorithm (ACO) [11], Particle Swarm Optimization (PSO) [12–14], and so on. Among them, PSO stands out for its ease of implementation and fast convergence. Representative ones include bare bones PSO (BBPSO) [15], evolutionary multitasking PSO (PSO-EMT) [16], evolve PSO (EPSO) [17], and so on.

However, PSO-based FS algorithms have three main shortcomings when solving FS tasks. Firstly, they can easily lose the diversity of population when dealing with high-dimensional datasets, which may cause the problem of falling into local optimum solutions. Secondly, most of them have not fully exploited the correlation information among features, which may lead to the premature convergence of PSOs. Finally, the learning exemplars of particles are very limited since each particle learns knowledge merely from its historical best position ($Pbest$) and the global best position ($Gbest$), which may reduce the likelihood of finding the optimal solution. In response to these issues, Tran et al. proposed a PSO-based FS algorithm with variable lengths (VLPSO) [18], which can provide a smaller and more promising subspaces for particles. Song et al. proposed a mutual information binary PSO algorithm (MIBBPSO) for exploiting the correlation information of features [15], which can effectively remove redundant (or irrelevant) features. Qu et al. proposed a size-adaptive PSO algorithm [19], which incorporates the importances of features into the population initialization, to enhance the diversity of population. Although these studies have made some progress, there is still much room for improvement in solving FS tasks.

1.1. Objective and contributions

Considering the above three aspects that needed to be improved, we propose a leader-adaptive particle swarm optimization with dimensionality reduction strategy (LAPSO-DR) for feature selection. The main contributions of this paper are as follows:

(1) A hybrid initialization strategy based on symmetric uncertainty (SU) is designed to obtain a diverse population. In particular, the features are ranked based on their SU values, and the population is divided into two parts at initial stage. Particles in the one part are initialized in different initialization ranges, while the rest particles are initialized in the usual way.
(2) A leader-adaptive strategy is proposed to enhance the exploitation ability of the population. Based on the strategy, each particle can have a different learning exemplar selected from the elite sub-swarm randomly, which is adaptively controlled by a dynamic regulation mechanism.
(3) A new inter-particle learning strategy is introduced to improve the exploration ability of the population. In this strategy, different dimensions of each particle can learn from different particles. What is more, the learning probabilities of particles are constrained so that poorer ones have a greater chance of becoming better.
(4) The dimensionality reduction strategy based on Markov blanket is introduced to reduce the size of the optimal feature subset. To save the computation, only the top thirty percent features of the elite particles are considered for finding essential features, which is more conducive to removing redundant features.

Next, the article is organized as follows. Section 2 provides an introduction to the related work. A detailed description of LAPSO-DR is given in Section 3. The design of the experiments and the analysis of the results are presented in Section 4. Finally, Section 5 is a summary of the whole article.

2. Background and related work

This part mainly reviews the concept of FS, as well as some classical PSO algorithms and their applications in FS tasks. Some EAs-based FS algorithms are also reviewed. In addition, the application of Markov blanket in FS is also introduced.

2.1. Feature selection

Suppose there is a dataset S with N examples and D features. FS is the process to select a subset X containing $d\,(d \le D)$ features from the dataset S such that the classification error rate $F(\cdot)$ is as small as possible. Usually, binary coding is used to represent the solution in FS task. A FS problem can be expressed as:

$$\begin{aligned} &\min F(X) \\ &s.t. X = (x_1, x_2, \dots, x_D), x_k \in \{0, 1\} \end{aligned} \tag{1}$$

where $x_k = 1$ indicates that the kth feature of X should be selected, otherwise not.

2.2. Particle swarm optimization

PSO was first used to solve the continuous optimization problems [20]. By simulating the behavior of the population, particles can find the optimal solution. When PSO is used for FS problems, the dimensions of the original dataset are the search space of particles, and each particle in the population represents a different solution. In the population, each particle learns from its historical best position ($Pbest$) and the global best position ($Gbest$). When the number of the current iteration is t, the velocity and position vectors of the ith particle are updated as follows [21]:

$$v_i^d = \omega v_i^d + c_1 r_1 (Pb_i^d - x_i^d) + c_1 r_2 (Gb^d - x_i^d) \tag{2}$$

$$x_i^d = x_i^d + v_i^d \tag{3}$$

where d represents one of the total features; ω is the inertia weight; c_1 and c_2 are acceleration constants; r_1 and r_2 are random numbers ($r_1, r_2 \in [0, 1]$). Generally, the value of x_i^d also represents the probability to select the dth feature of the ith particle.

When applying PSO into FS tasks, the position vectors of particles should be transformed into candidate feature subsets. In addition, we need to set a decoding threshold. If the value of x_i^d exceeds the threshold, the dth feature is selected; otherwise it is abandoned. The decoding threshold is set as 0.5 in our work.

2.3. PSO-based FS methods

PSO-based FS algorithms have made great progress in recent years. Researchers have combined PSO with other strategies to make them more suitable for FS tasks [22–24]. Some common strategies are initialization improvement, learning model adjustment, local search strategy, and so on.

Population initialization is one of the important steps in PSO-based FS algorithms, and an effective initialization strategy is conducive to improving the diversity of the population and the performance of algorithms [15,25–27]. For example, Gao et al. proposed an initialization strategy based on feature grouping [25], in which the features are divided into three groups according to the information gain rate. Each group is equipped with a different initialization strategy. This strategy can effectively remove some redundant features at the initial stage. Song et al. proposed a feature importance-based initialization method [15], in which the features of some particles in the population are removed or retained by the probability of correlation between features and labels. This method not only improves the diversity of the population, but also eliminates some redundant features at the initialization stage. What is more, Djellali et al. proposed a chaotic initialization method, in which the population is initialized by chaotic mapping [26]. Compared to random initialization, this method is able to increase the diversity of the population more. Xue et al. proposed a combination initialization strategy based on the filter [27], which can effectively improve the efficiency of the algorithm.

Usually, each particle in a population improves its quality by learning from its $Pbest$ and the $Gbest$. However, a single learning mode can easily make the population fall into a local optimum. To address this problem, Tran et al. proposed a new learning mechanism in the VLPSO [18]. This mechanism discards the $Gbest$ in the learning paradigm and gives each particle a chance to learn from other particles and its $Pbest$. This improvement can make particles to jump out of the local optimum easily. In addition, the constrained evolutionary mechanism proposed by Nguyen et al. [28] allows particles to choose a learning exemplar through a competitive mechanism. In particular, the learning exemplars are constrained to increase the probability of the particles for learning from the better exemplars.

To further improve the performance of PSO in FS tasks, many researchers have incorporated various local search strategies into the PSO algorithm. For example, Chen et al. developed a hybrid PSO based on the spiral-shaped mechanism (HPSO-SSM) [29], which can effectively enhance the exploitation ability of particles . Nguyen et al. proposed a sampling local search strategy [30], which uses the experience of previous evolution to improve the $Gbest$. In addition, Juhini et al. designed a flipped local search strategy [31]. This strategy is mainly used to improve the quality of $Gbest$ and help the population to jump out of the local optima. Moradi et al. proposed a local search strategy [32] with the "$Add(Delete)$" operator to enhance the search ability of PSO through adding (deleting) operation.

2.4. Review of other EAs-based FS methods

In addition to PSO, many evolutionary algorithms (EAs) have been applied into FS tasks and achieved good results [33–35]. In literature [36], Paniri et al. proposed a multi-label association-redundancy FS method based on ant colony algorithm (MLACO). In MLACO, the unsupervised and supervised heuristic functions are employed to find more relevant features, which provides a new idea to solve the multi-label FS problem. Hashemi et al. integrated a variety of heuristic ideas into ACO, and proposed a new ACO-based FS method (Ant-MCDM) [37], which can obtain useful information from multiple aspects instead of a single aspect. Zhou et al. proposed a non-dominated sorting genetic algorithm (PS-NSGA) [38] for multi-objective FS tasks. In order to improve the survival rate of elite particles, the precision-first dominant operator and the fast bit mutation operation are employed to improve its efficiency. Furthermore, the selection strategy is reasonably used to make PS-NSGA select the best feature subset. Based on the adaptive capuchin search algorithm (CSA), Braik et al. proposed three variants of CSA for FS task [39]. The improved variants of CSA adaptively adjust the influence factors such as inertia coefficient and acceleration coefficient, which makes their performance better than the initial version. Zhao et al. proposed a new binary dandelion algorithm for the FS task (SBDA) [40]. The SBDA improves the seeding strategy based on the standard BDA, and uses the chaos strategy to improve the search ability and diversity of the population, so it can better solve the FS problem. Inspired by the behavior of rat swarm, Awadallah et al. proposed a new FS method based on rat swarm optimization (BERSOC) [41]. In BERSOC, RSO is transformed into BRSO by the S-shape function, and the local search strategy of PSO and the crossover mechanism are employed in BRSO. In this way, the proposed BERSOC achieves good results on FS taks. Besides, a binary-based artificial rabbit optimization algorithm (ARO) was proposed to solve the FS task on medical datasets [42]. Hu et al. combined the reinforcement learning with GWO algorithm and proposed a new FS method (RLCGWO) [43]. In RLCGWO, the reinforcement learning model is established to obtain more valuable information accurately, with the purpose of improving its performance on FS tasks. Due to the limited space, the relevant literature cannot be all listed.

2.5. Markov Blanket in feature selection

The concept of Markov Blanket (MB) was first proposed by Judea Pearl in 1988. And then, MB has been used as an effective technique for FS tasks due to it can effectively identify redundant information in a dataset [44–46]. However, MB-based FS algorithms are always too time-consuming in finding the MB set for all variables, especially on high-dimensional datasets.

To address this issue, the concept of approximate Markov blankets (AMB) was proposed subsequently [47]. Compared with MB, AMB can save a lot of computation time by replacing a feature subset with a single feature, so it has been widely used in FS tasks [48–50]. For instance, Zhu et al. proposed a gene selection algorithm [48], which combines AMB with GA based on their advantage. In this algorithm, the MB-based memetic operators are proposed to make a good effect by adding (eliminating) the relevant (redundant) genes of the solution. For eliminating the misjudgment of strongly correlated features in AMB, Hua et al. restricted the judgment conditions of features in literature [49]. This method can effectively avoid the deletion of the features with high correlation, so as to further improve its classification accuracy. Zhou et al. embedded AMB as a local search strategy in GA algorithm [50]. After each iteration, AMB-based local search operator is employed to exploit the population. Due to the limited space, the relevant literature cannot be all listed.

3. The proposed method

To improve the performance of the PSO-based algorithm in FS tasks, a leader-adaptive particle swarm optimization with dimensionality reduction strategy (LAPSO-DR) is proposed in this paper. Compared with previous PSO-based FS algorithms, LAPSO-DR has four improvements, which are hybrid initialization strategy based on symmetric uncertainty (SU), the leader-adaptive strategy, the new inter-particle learning strategy, and the dimensionality reduction strategy based on Markov blanket. In the leader-adaptive strategy, each particle can have a different learning exemplar selected from the elite sub-swarm randomly, which is adaptively controlled by a dynamic regulation mechanism. Furthermore, the dimensionality reduction strategy based on Markov blanket can efficiently reduce the size of the optimal feature subset. Based on these two main reasons, the new proposed PSO-based algorithm is called leader-adaptive particle swarm optimization with dimensionality reduction strategy (LAPSO-DR). Fig. 1 shows the flowchart of the LAPSO-DR algorithm. These improvements are highlighted in the blue text box of the flowchart.

In the following section, the hybrid initialization strategy of LAPSO-DR is first introduced. And then, the new inter-particle learning strategy and the leader-adaptive strategy are described in detail. In addition, the dimensionality reduction strategy based on Markov Blanket is introduced. Finally, the whole process of LAPSO-DR algorithm is given.

3.1. Hybrid initialization strategy

To preliminary evaluate the qualities of features, the symmetric uncertainty (SU) [51] is used in our work. SU is a standardized version of the informative gain (IG), which can be employed to effectively evaluate the feature relevance in FS methods [52,53], as shown in Eq. (4). It is used to calculate the SU values of features and class labels, aiming at measuring the correlation between them. The larger the SU value, the stronger the correlation between them.

$$SU(X,Y) = \frac{2 \times IG(X,Y)}{H(X) + X(Y)} \tag{4}$$

where $IG(X,Y) = H(X) - H(X|Y)$ is the information gain, meaning the degree of uncertainty X is reduced when Y is known. $H(X)$ and $H(Y)$ are entropies of X and Y, respectively. By replacing X (Y) with features (labels), their SU values can be computed, and the same is true for the SU values among features.

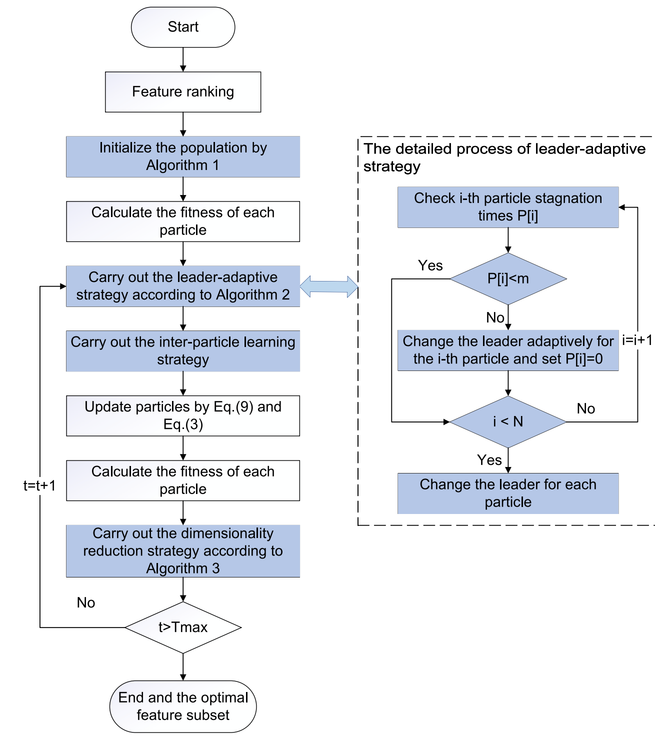

Fig. 1. The framework of LAPSO-DR.

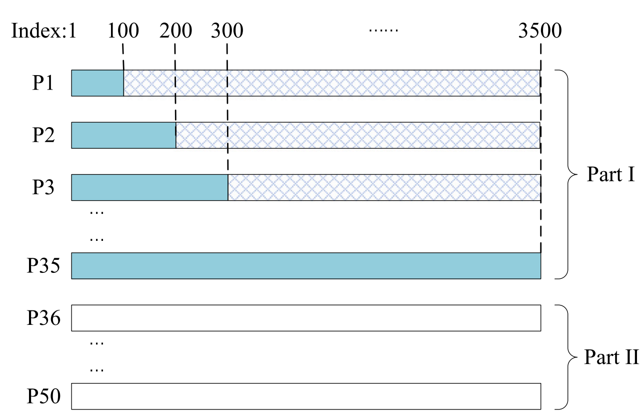

Fig. 2. An example of population initialization: the size of population is 50, the number of features is 3500, and $DivRatio = 7 : 3$.

In this paper, we propose a new hybrid initialization strategy based on feature rearrangement, aiming to obtain a diverse population. We divide the population into two parts, and set the division ratio of the population to be $DivRatio$. Particles in the first part have different initialization ranges, as shown in Eq. (5). The rest particles are initialized randomly in the whole search space.

$$Len(X_i) = \left\lceil \frac{i \times D}{N} \right\rceil \tag{5}$$

where i represents the sequence number of particles; D is the total number of features; N is the size of the population. $Len(X_i)$ is the initialization range of ith particle. Fig. 2 shows a schematic of this hybrid initialization strategy. Algorithm 1 shows the pseudocode of the initialization strategy.

3.2. The new inter-particle learning strategy

To improve the exploration ability of the population, we introduce a new inter-particle learning strategy based on CLPSO [54] and VLPSO [18]. In this strategy, each particle has a certain probability, which is based on its sequence number, to learn from the $Pbests$ of other particles. Different from CLPSO, the learning probability P is reset for ranking the fitness of a particle in each iteration, as shown in Eq. (6). In this way, the worse particle can have a higher probability for learning from other exemplars. In addition, the other settings of the proposed learning strategy are the same as CLPSO.

$$P_i = 0.05 + 0.45 * \frac{exp^{\frac{10(Rank(i)-1)}{N-1}}}{exp^{10} - 1} \tag{6}$$

where N is the size of the population; $Rank(i)$ is the rank of the ith particle. The best particle in the population is ranked as 1.

3.3. The leader-adaptive strategy

To enhance the exploitation ability of the population, we introduce a leader-adaptive strategy in LAPSO-DR. In this strategy, a certain number of particles with high quality are selected to form the elite

Algorithm 1 The hybrid initialization strategy

Input: The dataset S, the population size N, the division ratio of the population $DivRatio$, the size of first part in the population $Fsize$, and the total number of features D.
Output: A initialized population X.

Calculate the SU value $SU(F_i, C)$ for each feature F_i based on the dataset S;
Rearrange these features in descending order based on their SU values;
Calculate the length of each particle $Len(X_i)$ in the first part in the population according to Eq. (5);
for $i = 1$ to $Fsize$ **do**
  for $d = 1$ to $Len(X_i)$ **do**
    $X_{id} = rand$;
  end for
end for
for $i = Fsize + 1$ to N **do**
  for $d = 1$ to D **do**
    $X_{id} = rand$;
  end for
end for
return X;

sub-swarm, and then each particle randomly selects its leader from the elite sub-swarm. At the same time, a threshold m is defined to determine adaptively whether the particle must to change its leader particle. When the number of stagnations of a particle exceeds m, the particle will choose its leader exemplar from the elite sub-swarm again.

To ensure the superiority of the elite particles, we set the size of the elite sub-swarm within a reasonable range. In the early stage of evolution, the size of the elite sub-swarm is set to be $0.5 * N$, aiming to improve the diversity of the population. In the medium term, the size of the elite sub-swarm becomes smaller gradually when the number of iterations grows, as shown in Eqs. (7) and (8). In the later stage, the size of the elite sub-swarm reaches a minimum value ($0.1 * N$), with the purpose of speeding up the convergence. When the number of the current iteration is t, the new updating formula for the velocity vector is shown in Eq. (9).

$$h = -0.4 * \frac{t}{T} + 0.5, h \in [0.1, 0.5] \tag{7}$$

$$ESize = h * N \tag{8}$$

$$v_i^d = \omega v_i^d + c_1 r_1 (Pb_i^d - x_i^d) + c_1 r_2 (EL_i^d - x_i^d) \tag{9}$$

where h is the control parameter that decreases linearly with the number of iterations; $ESize$ is the size of the elite sub-swarm; T is the total number of iterations. EL saves the elite particle learned by each particle. Algorithm 2 describes the leader-adaptive strategy, in which particles can select different leaders.

3.4. The dimensionality reduction strategy based on Markov blanket

To reduce the size of the optimal feature subset, the dimensionality reduction strategy based on Markov blanket is employed on elite sub-swarm. In particular, the AMB-based method is adopt to add (delete) strongly correlated (redundant) features in the elite particles. In this way, the quality of the elite sub-swarm can be further improved. The concepts of MB and AMB are described in detail as follows:

Definition 1. (*Markov Blanket*) Suppose there is a feature set S, a class label C, and a feature subset $Q \subseteq S$. And there exists a feature F_i that belongs to S but not to Q. Given Q, if $(S \cup C) - Q - F_i$ is conditionally independent, then Q is an MB of F_i.

Algorithm 2 The leader-adaptive strategy

Input: The population X, the size of population N, the leaders of particles EL, the size of the elite sub-swarm $ESize$, the number of stagnation times of each particle $flag$, and the threshold m;
Output: The updated EL, the updated $flag$

Calculate the fitness values of X, the top $ESize$ ranked particles form the elite sub-swarm;
for $i = 1$ to N **do**
  if $flag(i) > m$ **then**
    Reselect a leader particle randomly from the elite sub-swarm for X_i and update EL_i;
    $flag(i) = 0$;
  end if
end for
return EL, $flag$;

This means that for the class label C, the information provided by Q alone can contain the information provided by F_i, so F_i can be regarded as a redundant feature that should be removed. In general, the MB-based methods have a high computational complexity.

Definition 2 (*Approximate Markov Blanket*)**.** For any two features $F_i, F_j \in S, (i \neq j)$, if $SU(F_i, C) \geq SU(F_j, C)$ and $SU(F_i, F_j) \geq SU(F_j, C)$, then F_i is the AMB of F_j.

To further reduce the computational complexity, we perform the AMB on the top 30% of the features of the candidate elite particles. By doing this, the quality of particles in the population can be improved effectively. The detailed process is shown in Algorithm 3.

Algorithm 3 The dimensionality reduction strategy based on MB

Input: The elite sub-swarm E, the total number of features D, the size of the elite sub-swarm $LSize$;
Output: The new elite sub-swarm E';

Divide the top $0.3 * D$ features of the elite particles into two groups, Q_1 and Q_0. The features with the values greater than 0.5 will be assigned to Q_1 and the remaining will be assigned to Q_0;
Calculate the average SU value of Q_1, denoted as $AvgSU$;
while $i < LSize$ **do**
  for $d = 1$ to $size(Q_1)$ **do**
    for $j = d + 1$ to $size(Q_1)$ **do**
      if $SU(F(j), F(d)) \geq SU(F(j), C)$ **then**
        Set $F(j) = 0$ and remove the j-th feature from E_i;
      end if
    end for
  end for
  for $k = 1$ to $size(Q_0)$ **do**
    if $SU(F(k), C) > AvgSU$ **then**
      Set $F(k) = 1$ and add the k-th feature to E_i;
    end if
  end for
  Calculate the fitness of the new E_i;
  if fit(new E_i) is better than fit(E_i) **then**
    Change E_i to new E_i;
  end if
end while
return E';

3.5. The framework of LAPSO-DR

By incorporating the aforementioned components, the pseudocode of LAPSO-DR is shown in Algorithm 4, and the flowchart of it is sketched in Fig. 1.

Algorithm 4 LAPSO-DR

Input: Training dataset S;
Output: The optimal feature subset;

```
Rearrange features by SU(F, C) values;
Initialize the population according to algorithm 1;
Calculate the fitness of the population according to Eq. (10);
while termination iteration is not reached do
    Calculate the size ESize of the elite sub-swarm by Eq. (7) and Eq. (8);
    The top ESize best particles form the elite sub-swarm;
    Use the leader-adaptive strategy according to algorithm 2;
    Use the new inter-particle learning strategy;
    Update the velocity and position vectors by Eq. (9) and Eq. (3);
    Calculate the fitness of the updated population according to Eq. (10);
    Conduct the dimensionality reduction strategy on the elite sub-swarm according to algorithm 3;
    Update the elite sub-swarm;
    Update Gbest if it changes;
end while
return the optimal feature subset;
```

Table 1
Basic information of 18 datasets.

Datasets	#Features	#Instances	#Classes
Segmentation	19	2310	17
WDBC	30	569	2
Ionosphere	34	351	2
Sonar	60	208	2
Hill-valley	101	505	2
Musk1	166	476	2
Divorce	170	54	2
Arrhythmia	279	452	16
LSVT	310	126	2
Madelon	500	2600	2
Ioslet5	618	1559	26
MFS	649	2000	10
ORL	1024	400	40
SRBCT	2308	83	4
DLBCL	5469	77	2
Leukemia1	5327	72	3
Leukemia2	7129	72	4
Leukemia3	11 225	72	3

3.6. Time complexity analysis

The time complexity of LAPSO-DR consists of five aspects. (A) the computation of SU values between features and labels, feature ranking; (B) initialization process; (C) fitness evaluation process; (D) the leader-adaptive strategy; (E) the dimensionality reduction process based on MB. For part A, it needs $O(D)$ to compute the SU values of features and labels, and needs $O(D \times log(D))$ for feature ranking. Therefore, this part costs $O(D + D \times log(D))$, where D is the total number of features. For part B, the initialization process costs $O(N \times D)$, where N is the size of the population. Part C is determined by the specific task, so that its time complexity is not considered here. In part D, the elite particles need to be selected based on their qualities, which costs $O(N \times log(N))$. Finally, the dimensionality reduction process has two operations: deleting and adding features. The deleting operation requires $O(M \times dlog(d))$, while the adding operation requires $O(N)$, where M is the size of the elite sub-swarm and d is $0.3 * D$. Based on this analysis, part E costs $O(N + M \times dlog(d))$. What is more, parts D and E participate in the iterative loop process with a number of iterations T. Therefore, the time complexity of LAPSO-DR is $O(ND + Dlog(D) + TNlog(N) + TM \times dlog(d))$ in total.

4. Experimental studies

In this section, the effectiveness of the proposed LAPSO-DR is validated. First, we present the benchmark datasets, comparison algorithms, and their parameter settings in detailed. Furthermore, the fitness function is introduced. Moreover, experimental results obtained by the proposed algorithm and the comparison algorithm are compared and analyzed. Finally, experiments analysis of parameter sensitivity and three main strategies are shown.

4.1. Benchmark datasets

In our work, 18 benchmark datasets are selected to validate the performance of the LAPSO-DR. These datasets include low-dimensional datasets (less than 100 features), medium-dimensional datasets (number of features between 100 and 500), and high-dimensional datasets (number of features over 500). Table 1 shows the detail information of these 18 datasets, which can be found in the UCI machine learning repository (http://archive.ics.uci.edu/ml) and the open sources (https://ckzixf.github.io/dataset.html).

To reduce bias of experimental results, we randomly split each dataset into two parts at a ratio of $7 : 3$, with 70% as the training set and 30% as the test set. KNN is used as the evaluation classifier in all algorithms, in which the k-value is set to be 1. During training stage, tenfold cross-validation is used as internal evaluation. In addition, to illustrate the generality of LAPSO-DR and to analyze its performance on different classifiers, we also compare LAPSO-DR with other 8 algorithms using SVM as a classifier. For fairness concerning, each algorithm is run 30 times independently on each dataset, and the average value of the 30 results is used for comparison.

4.2. Comparison algorithms and parameter setting

To evaluate the performance of LAPSO-DR, we compare it with 8 state-of-art wrapper-based FS algorithms. They are as follows.

- A novel FS method based on PSO and MFO (PSO-EMT) [55]
- A pyramid PSO algorithm with a new competitive and cooperative strategy (PPSO) [56]
- Hybrid PSO with the spiral-shape mechanism (HPSO-SSM) [29]
- A binary version of hybrid PSOGWO (BGWOPSO) [57]
- Competitive swarm optimizer (CSO) [58]
- Binary BBPSO algorithm (Binary BBPSO) [59]
- The standard PSO algorithm (PSO)
- The genetic algorithm (GA)

For all algorithms, the size of the population is set to be 50, and the maximum number of iterations is set to be 100. The detailed parameter settings and computational complexity of each comparison algorithm are shown in Table 2. All experiments were conducted on PC with Intel Core I7–12700 CPU @2.10 GHz. on MATLAB 9.13 (R2022b).

4.3. Fitness function

In the FS task, there are two important goals: minimizing the error rate of classification and minimizing the size of the optimal feature subset. Therefore, we combine the error rate of the classification with the size of selected feature subset in the fitness function, as shown in Eq. (10).

$$fitness = \alpha * Error + (1 - \alpha) * \frac{Size(subset)}{D} \tag{10}$$

where $Error$ is the error rate of classification; $Size(subset)$ is the size of selected feature subset; D is the total number of features. α is the weight parameter, which is used to control the importance between the error rate and the size of selected feature subset. Considering the former is more important than the later in FS task, α is set as 0.9 in our work.

Table 2
Parameter settings of all algorithms.

Algorithms	Parameter settings	Computational complexity
PSO-EMT	$c_1 = c_2 = 1.49445, w = 0.9 - 0.5 * (t/T), rmp = 0.6$	Approximately $O(N_1 D_1 T + (N + N_1)D)$
PPSO	$Layers = [4, 8, 20, 32], \rho = 0.4$	Approximately $O(N log(N) + N + D + T N D)$
HPSO-SSM	$c_1 = c_2 = 2, w = 0.9 - 0.5 * (t/T), p = 0.8$	Approximately $O((T + M_1) \times (N + M_2) \times Q)$
BGWOPSO	$c_1 = c_2 = 1.49445, w = 0.9 - 0.5 * (t/T)$	Approximately $O(T N_1 \times D + N D)$
CSO	$\phi = 0.1, \lambda = 0.5$	$O(\frac{1}{2} \times T \times N \times D)$
BBPSO	$c_1 = c_2 = 2,\ w = 1$	Approximately $O(N + N D + N_1 DT)$
PSO	$c_1 = c_2 = 1.49445, w = 0.9 - 0.5 * (t/T)$	$O(T \times N \times D)$
GA	$pm = 0.001, pc = 1$	Approximately $O(T \times N \times D)$
LAPSO-DR	$c_1 = c_2 = 1.49445, w = 0.9 - 0.5 * (t/T), m = 3$	$O(N D + Dlog(D) + T N log(N) + T M \times dlog(d))$

Table 3
LAPSO-DR is compared with the other eight algorithms in terms of classification accuracy (by KNN).

Dataset	GA	PSO	BBPSO	CSO	BGWOPSO	HPSO-SSM	PPSO	PSO-EMT	LAPSO-DR
Segmentation	96.70	96.56	96.67	96.52	86.33	**96.95**	86.85	90.13	96.76
WDBC	94.68	94.66	94.37	94.78	94.70	94.89	94.52	94.25	**94.97**
Ionosphere	89.97	87.84	89.02	88.76	88.76	89.49	87.33	82.76	**93.14**
Sonar	83.44	84.14	83.39	84.30	84.84	84.25	84.41	77.63	**93.28**
Hill-Valley	53.77	54.85	54.42	54.89	54.03	52.64	54.40	52.42	**60.26**
Musk1	86.04	87.13	87.18	86.06	86.25	86.27	87.76	88.11	**92.63**
Divorce	97.71	97.65	97.12	97.52	97.12	96.14	96.14	96.79	**98.63**
Arrhythmia	53.87	54.9	56.27	55.25	54.85	56.54	57.01	49.95	**63.53**
LSVT	77.28	77.19	78.07	78.25	78.77	76.49	78.95	82.46	**90.96**
Madelon	56.71	57.54	57.08	57.75	56.39	84.93	58.23	88.12	**88.97**
Isolet5	83.38	83.8	83.63	83.7	83.35	81.99	85.16	64.59	**88.53**
MFS	97.68	97.71	97.54	97.70	97.69	97.34	97.74	84.44	**98.24**
ORL32	90.47	90.56	88.72	90.17	90.53	89.81	89.94	86.08	**91.44**
SRBCT	83.07	80.53	82.27	84.40	83.07	85.60	81.73	95.20	**95.33**
DLBCL	83.33	83.91	81.59	84.06	83.77	83.91	84.35	94.20	**94.35**
Leukemia1	79.09	85.91	85.45	84.35	84.55	73.18	77.73	90.45	**93.18**
Leukemia2	77.73	90.91	78.18	87.88	87.73	87.73	79.09	97.77	**98.64**
Leukemia3	84.55	88.18	87.73	85.91	85.00	85.91	87.73	91.82	**92.73**

4.4. Experimental results and analysis

4.4.1. Comparison between LAPSO-DR and competitors

In this section, we present a comparative analysis on experimental results obtained by LAPSO-DR and the other eight algorithms. The first part is a comparison of the experimental performance of LAPSO-DR with other competitors using KNN simultaneously. From the Table 3, it can be seen that the LAPSO-DR achieves higher classification accuracies on low-dimensional datasets, except for Segmentation dataset. On Segmentation, HPSO-SSM has the highest classification accuracy. Especially, on the Sonar and Ionosphere datasets, the classification accuracies obtained by LAPSO-DR are much higher than those of competitors. For the medium-dimensional datasets Hill-Valley, Musk1, Divorce, Arrhythmia, and LSVT, LAPSO-DR achieves the best classification accuracies. On the high-dimensional datasets, LAPSO-DR achieves higher classification accuracies than its competitors. In particular, the classification accuracy obtained by LAPSO-DR reaches 95.33% (94.35%) on the SRBCT (DLBCL) datasets. There are little difference in classification accuracy between LAPSO-DR and PSO-EMT on the SRBCT and DLBCL datasets.

Table 4 lists the sizes of the selected feature subsets achieved by LAPSO-DR and competitors. On the low-dimensional datasets, LAPSO-DR ranks second only to PSO-EMT and HPSO-SSM in terms of the size of selected feature subset. In particular, PSO-EMT achieves the smallest size of selected feature subset on the low-dimensional datasets except for the Segmentation dataset. HPSO-SSM obtains the smallest size of feature subset on the Sementation dataset. For the medium-dimensional datasets, the size of the optimal feature subset obtained by LAPSO-DR is only larger than that of PSO-EMT. At the same time, PSO-EMT obtains the smallest sizes of selected feature subsets on the Hill-Valley, Divorce, Arrhythmia, and LSVT datasets. On high-dimensional datasets, LAPSO-DR achieves the smallest size of selected feature subset only on the Madelon dataset. On the Isolet5, MFS, ORL32, SRBCT, DLBCL, Leukemia1, Leukemia2, and Leukemia3 datasets, the sizes of the optimal feature subsets obtained by LAPSO-DR are slightly larger than those of PSO-EMT, but smaller than those of the other seven competitors.

Figs. 3 and 4 show the classification accuracies and the sizes of the optimal feature subsets obtained by LAPSO-DR and other 8 algorithms on 18 datasets, respectively. Be noted that, an axis represents a dataset and a closed line represents an algorithm in Figs. 3 and 4. In Fig. 3, the point on the axis that is farther from the center indicates a higher classification accuracy obtained by the algorithm on that dataset. In Fig. 4, the point on the axis that is closer to the center means that the size of selected feature subset obtained by the algorithm is smaller. Obviously, from Fig. 3, it can be seen that the points of LAPSO-DR are farthest from the center on most axes, indicating that the classification accuracies achieved by LAPSO-DR are better than those of competitors on most datasets. From Fig. 4, it can be seen that the points of LAPSO-DR are closer to the center on most axes, which shows that the sizes of the optimal feature subsets obtained by LAPSO-DR are smaller.

Tables 5 and 6 list the comparison between LAPSO-DR and competitors in terms of the average running time and the average fitness, respectively. For the average running time, LAPSO-DR is better than most of these competitors, but it is slightly more time-consuming than two algorithms (PPSO, PSO-EMT). On low-dimensional datasets, the average running times achieved by these algorithms are similar except for PPSO. On medium-dimensional and high-dimensional datasets, LAPSO-DR has longer running time on Madelon, Isolet5, and MFS datasets, and the average running times on the other datasets are close to HPSO-SSM and BGWOPSO. In general, LAPSO-DR has little difference with most of competitors in the average running time. Compared with other algorithms, LAPSO-DR contains four strategies and the source of extra time-consuming is the dimensionality reduction strategy based on Markov blanket. However, according to the performance improvement brought by this strategy for LAPSO-DR in Table 8, this part of time consumption is worth it.

Table 4
LAPSO-DR is compared with the other eight algorithms in terms of the size of obtained feature subset (by KNN).

Dataset	GA	PSO	BBPSO	CSO	BGWOPSO	HPSO-SSM	PPSO	PSO-EMT	LAPSO-DR
Segmentation	3.23	3.90	3.20	3.57	14.03	**3**	11.36	4.93	3.06
WDBC	7.63	8.23	7.10	7.87	8.57	8.80	6.03	**3.03**	4.53
Ionosphere	8.70	9.87	11.70	10.03	10.77	6.73	6.37	**2.03**	7.93
Sonar	25.57	25.03	28.33	25.27	31.00	16.3	22.77	**5.43**	27.06
Hill-Valley	42.13	39.93	45.23	39.83	51.00	6.10	35.33	**2.93**	28.97
Musk1	73.10	72.80	80.83	73.70	97.26	**31**	65.10	63.33	71.10
Divorce	14.33	10.03	8.70	11.03	7.30	2.93	3.53	**2.50**	4.57
Arrhythmia	127.97	131.57	133.97	182.87	182.87	20.87	120.67	**7.50**	91.93
LSVT	147.90	140.37	147.17	147.00	203.63	40.07	145.47	**19.36**	96.93
Madelon	241.23	238.17	234.17	139.53	341.20	17.53	224.80	10.63	**9.70**
Isolet5	294.83	287.77	288.97	292.77	399.83	81.06	269.57	**35.03**	183.37
MFS	281.70	272.43	256.13	272.73	324.77	141.30	219.20	**16.63**	39.97
ORL32	491.60	480.90	498.80	480.56	614.40	147.70	461.03	**41.37**	221.9
SRBCT	1145.70	1113.70	1114.00	356.67	1467.40	155.43	1113.00	**37.20**	182.93
DLBCL	2717.90	2660.30	2654.90	616.43	3334.70	308.53	2688.80	**85.87**	278.03
Leukemia1	2646.70	1784.00	2592.50	793.10	3334.90	404.50	2615.90	**64.40**	99.50
Leukemia2	3574.70	2294.80	3518.20	456.97	4353.10	642.70	3520.70	**26.50**	88.60
Leukemia3	5554.00	3423.50	5446.30	741.73	6715.40	685.30	5431.10	**221.30**	408.50

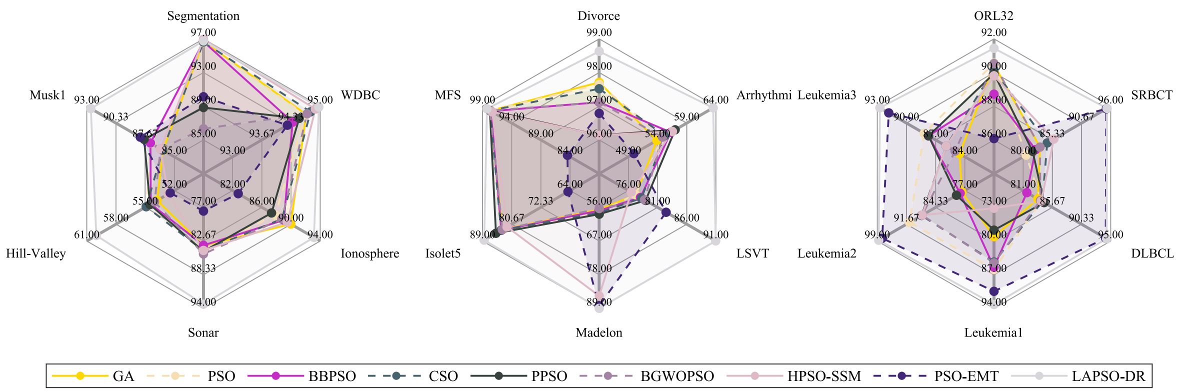

Fig. 3. The comparison of classification accuracies between LAPSO-DR and other eight algorithms.

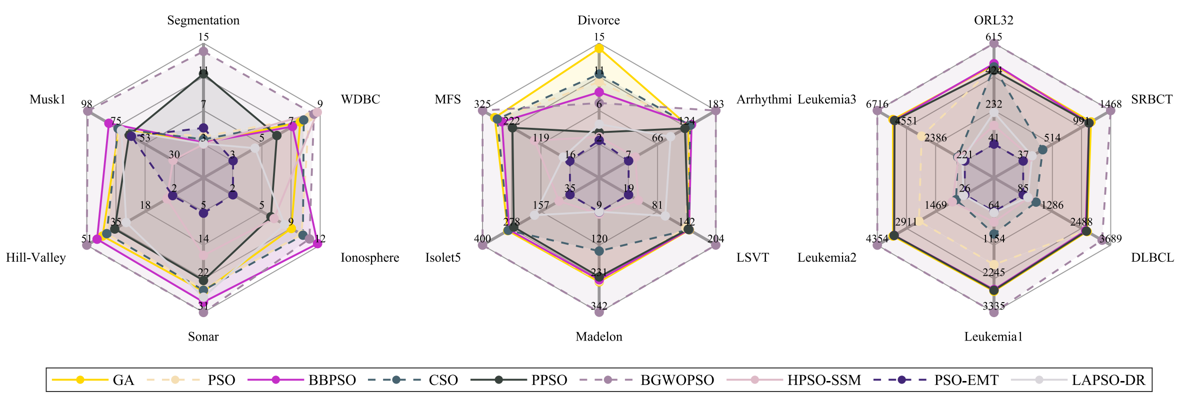

Fig. 4. The comparison of the sizes of selected feature subsets between LAPSO-DR and other eight algorithms.

The fitness value is used as an evaluation measure for the performance of algorithms. In this paper, the smaller the fitness value is, the better the comprehensive performance of the algorithm. On the low-dimensional datasets Segmentation, Ionosphere, and Sonar, the fitness values of LAPSO-DR are 0.038, 0.039, and 0.042, respectively. However, HPSO-SSM has the best fitness on the low-dimensional datasets WDBC and Hill-Valley. On the medium-dimensional dataset, the fitness values of LAPSO-DR are smaller than the other competitors except for the dataset Arrhythmia. At the same time, HPSO-SSM has better fitness value on dataset Arrhythmia. The fitness values obtained by LAPSO-DR are better than competitors on most high-dimensional datasets. However, on three high-dimensional datasets SRBCT, DLBCL, and Leukemia, the fitness values of LAPSO-DR are only greater than that of PSO-EMT. Taken together, the fitness values of LAPSO-DR are better than those of competitors on 15 out of 18 datasets, which indicates that LAPSO-DR is competitive in solving FS tasks.

In order to explore the effect of different classifiers on the experiments, the KNN and SVM classifier are employed to conduct comparison experiments between LAPSO-DR and competitors. The comparison results are shown in Tables 7 and 8.

Table 7 shows the experimental results obtained by LAPSO-DR and competitors in terms of the classification accuracy, where the SVM classifier was used by them. On the low-dimensional datasets, LAPSO-DR achieves the highest classification accuracies on Ionosphere, Sonar,

Table 5
LAPSO-DR is compared with the other eight algorithms in terms of the average running time (s) (by KNN).

Dataset	GA	PSO	BBPSO	CSO	BGWOPSO	HPSO-SSM	PPSO	PSO-EMT	LAPSO-DR
Segmentation	206.89	113.95	279.90	378.03	29.31	111.76	**27.66**	277.32	180.38
WDBC	13.17	18.72	**7.64**	85.67	15.18	51.80	12.21	21.87	39.81
Ionosphere	25.34	6.05	30.23	28.68	9.64	8.73	**4.25**	23.89	15.86
Sonar	17.02	6.10	13.19	21.24	21.04	6.99	**3.38**	6.27	15.24
Hill-valley	7.02	27.10	33.58	30.38	6.84	13.50	**4.16**	25.87	24.82
Musk1	78.82	63.73	**10.63**	45.89	28.03	43.33	48.77	76.58	52.06
Divorce	7.01	4.25	12.69	3.37	3.42	4.84	3.77	**3.24**	21.16
Arrhythmia	15.11	33.07	101.86	8.57	14.01	51.54	8.27	**5.28**	49.78
LSVT	4.10	26.95	21.48	**2.12**	5.25	52.01	2.24	6.36	12.69
Madelon	625.17	375.23	344.70	405.17	518.30	265.00	376.23	**176.14**	642.37
Ioslet5	268.43	199.07	164.71	169.14	216.58	234.94	218.16	**65.24**	562.60
MFS	402.73	300.44	269.74	279.28	264.75	349.97	349.10	**149.42**	304.14
ORL32	314.79	207.77	195.97	**19.12**	38.02	344.67	161.17	170.30	138.95
SRBCT	12.51	32.58	12.90	19.46	27.73	50.63	**8.87**	17.83	30.33
DLBCL	84.88	51.67	25.23	24.07	63.73	54.37	**13.99**	14.13	40.67
Leukemia1	84.29	81.55	78.60	25.53	56.57	13.54	13.07	**9.09**	53.66
Leukemia2	105.12	93.39	113.94	29.54	42.74	16.23	72.54	**9.39**	13.63
Leukemia3	142.82	93.44	160.23	45.87	115.50	19.42	23.11	**11.99**	71.50

Table 6
LAPSO-DR is compared with the other eight algorithms in terms of the average fitness (by KNN).

Dataset	GA	PSO	BBPSO	CSO	BGWOPSO	HPSO-SSM	PPSO	PSO-EMT	LAPSO-DR
Segmentation	0.041	0.041	0.043	0.040	0.136	0.039	0.105	0.092	**0.038**
WDBC	0.053	0.055	0.049	0.049	0.046	**0.025**	0.039	0.045	0.039
Ionosphere	0.104	0.102	0.077	0.092	0.113	0.072	0.084	0.045	**0.039**
Sonar	0.107	0.094	0.050	0.114	0.119	0.106	0.111	0.112	**0.042**
Hill-valley	0.445	0.387	0.357	0.385	0.438	**0.325**	0.396	0.351	0.332
Musk1	0.130	0.125	0.114	0.125	0.120	0.093	0.109	0.091	**0.038**
Divorce	0.026	0.028	0.027	0.026	0.020	0.009	0.015	0.002	**0.001**
Arrhythmia	0.516	0.488	0.449	0.501	0.469	**0.443**	0.462	0.552	0.459
LSVT	0.156	0.119	0.082	0.143	0.176	0.129	0.119	0.117	**0.070**
Madelon	0.408	0.396	0.384	0.401	0.414	0.147	0.388	0.104	**0.091**
Ioslet5	0.172	0.183	0.165	0.186	0.180	0.157	0.145	0.310	**0.107**
MFS	0.059	0.049	0.047	0.050	0.056	0.017	0.040	0.120	**0.012**
ORL32	0.109	0.099	0.067	0.093	0.104	0.070	0.102	0.083	**0.041**
SRBCT	0.117	0.133	0.114	0.047	0.139	0.028	0.102	**0.001**	0.022
DLBCL	0.115	0.126	0.128	0.121	0.148	0.074	0.118	**0.001**	0.022
Leukemia1	0.117	0.141	0.158	0.104	0.178	0.076	0.129	**0.017**	0.032
Leukemia2	0.171	0.098	0.202	0.010	0.123	0.052	0.129	0.003	**0.002**
Leukemia3	0.080	0.128	0.124	0.034	0.123	0.050	0.103	0.025	**0.019**

Table 7
LAPSO-DR is compared with the other eight algorithms in terms of classification accuracy (by SVM).

Dataset	GA	PSO	BBPSO	CSO	BGWOPSO	HPSO-SSM	PPSO	PSO-EMT	LAPSO-DR
Segmentation	92.98	96.90	96.01	96.54	90.86	**96.94**	96.68	90.13	96.25
WDBC	94.54	**95.91**	94.15	95.13	95.56	93.18	93.45	94.85	95.71
Ionosphere	87.62	87.62	86.67	86.35	84.76	87.43	87.43	82.86	**89.52**
Sonar	76.88	83.87	85.48	82.26	72.58	84.19	85.48	80.97	**87.42**
Hill-valley	50.65	57.14	56.71	56.71	52.38	51.52	54.55	54.29	**58.44**
Musk1	80.77	80.77	**91.14**	86.25	85.55	88.34	85.03	89.93	87.65
Divorce	96.73	96.73	95.42	97.39	97.65	95.42	97.65	97.39	**99.02**
Arrhythmia	57.11	52.21	59.56	56.62	56.86	56.47	57.35	51.32	**60.05**
LSVT	78.95	81.58	85.96	78.95	71.93	73.68	78.95	80.00	**90.79**
Madelon	57.48	56.35	69.36	74.32	57.74	87.78	59.00	87.67	**88.72**
Ioslet5	84.47	83.55	86.04	82.34	84.19	82.19	86.07	66.15	**91.28**
MFS	97.17	97.08	97.50	96.06	97.10	96.61	97.28	85.70	**97.75**
ORL32	89.72	89.17	88.06	88.61	88.06	87.78	88.33	87.17	**90.56**
SRBCT	80.00	89.33	82.67	76.00	81.33	86.67	89.60	**96.00**	**96.00**
DLBCL	78.26	86.96	85.51	86.96	91.30	91.30	82.61	94.78	**95.65**
Leukemia1	90.91	84.09	84.85	88.18	80.30	90.00	92.42	**94.55**	93.18
Leukemia2	90.91	88.64	81.82	86.36	81.82	83.33	89.39	91.82	**97.73**
Leukemia3	86.36	93.18	86.36	87.88	87.88	87.88	93.94	93.64	**95.45**

and Hill-Valley, but it achieves slightly lower classification accuracies than HPSO-SSM and PSO on Segmentation and WDBC datasets. On the medium-dimensional datasets, LAPSO-DR has a lower classification accuracies than BBPSO only on Musk1, and its classification accuracies are higher than competitors on the remaining datasets. Furthermore, the classification accuracies of LAPSO-DR are more than 90% on high-dimensional datasets. In particular, LAPSO-DR has a lower classification accuracies than PSO-EMT on Leukemia1, but it has a better classification accuracies than competitors on other high-dimensional datasets. The classification accuracies of LAPSO-DR on DLBCL, Leukemia1, Leukemia2, and Leukemia3 datasets reach 95.65%, 93.18%, 97.73%, and 95.45%, respectively. This proves that the overall accuracy of LAPSO-DR is still better than competitors when the SVM classifier is employed.

Table 8
LAPSO-DR is compared with the other eight algorithms in terms of the size of obtained feature subset (by SVM).

Dataset	GA	PSO	BBPSO	CSO	BGWOPSO	HPSO-SSM	PPSO	PSO-EMT	LAPSO-DR
Segmentation	5.00	4.00	**3.00**	4.00	5.00	3.40	3.20	4.40	**3.00**
WDBC	6.67	8.50	5.00	6.33	6.20	3.33	4.40	**3.00**	5.00
Ionosphere	10.33	11.50	8.00	7.00	12.00	4.20	7.20	**2.20**	6.50
Sonar	22.67	24.00	24.67	24.33	30.33	11.40	23.60	**6.80**	13.80
Hill-valley	33.67	42.00	41.00	42.33	29.33	5.67	36.80	**5.00**	12.00
Musk1	77.50	84.50	66.67	41.33	103.00	**26.67**	69.60	66.40	36.67
Divorce	14.00	14.33	6.00	10.33	7.00	**2.33**	4.40	3.33	2.50
Arrhythmia	125.33	136.50	123.67	52.00	27.00	118.20	181.00	**7.40**	51.33
LSVT	150.00	147.00	128.00	151.67	208.33	32.67	144.20	**21.00**	42.50
Madelon	238.00	244.50	221.33	132.67	336.33	**5.67**	224.80	12.80	9.00
Ioslet5	297.65	287.50	282.67	146.33	402.00	69.33	279.80	**37.60**	205.20
MFS	275.50	287.50	257.67	75.67	340.00	64.33	197.00	**19.40**	32.00
ORL32	486.67	501.00	478.00	194.00	631.33	132.67	482.80	**39.80**	236.00
SRBCT	1053.50	1062.70	998.00	97.67	1490.00	55.00	1142.80	**37.20**	84.50
DLBCL	2726.70	2596.00	2690.00	516.33	2664.30	**43.00**	2694.20	69.80	61.00
Leukemia1	2573.00	2582.00	2597.33	599.60	3211.30	362.00	2545.00	**78.60**	84.00
Leukemia2	3542.00	3460.00	3478.70	359.33	4220.70	448.57	3552.00	170.60	**71.00**
Leukemia3	5463.00	5461.50	5606.70	1126.30	6765.30	623.67	5492.00	**128.40**	312.50

Table 8 presents the comparison between LAPSO-DR and competitors in terms of the size of the obtained feature subset. On low-dimensional datasets, the size of feature subset obtained by LAPSO-DR is the smallest on Segmentation, while the sizes of feature subsets obtained by PSO-EMT are the smallest On the other datasets. In general, the sizes of selected feature subsets obtained by LAPSO-DR and competitors are less different on low-dimensional datasets. On the medium-dimensional datasets, the sizes of the feature subset obtained by LAPSO-DR are only larger than PSO-EMT, but smaller than the other seven competitors. For high-dimensional datasets, LAPSO-DR obtains the smallest size of feature subset on Leukemia2, and the rest of the smallest feature subsets are obtained by PSO-EMT algorithm. Obviously, the sizes of feature subset obtained by LAPSO-DR and PSO-EMT are much smaller than that of other competitors on high-dimensional datasets. It can be seen that LAPSO-DR is also competitive in feature elimination.

In summary, LAPSO-DR has a higher classification accuracy than the other eight algorithms. Furthermore, the performance of LAPSO-DR is second only to PSO-EMT in terms of the size of the selected feature subset.

4.4.2. Statistical analyses

To further analyze the significance difference between LAPSO-DR and competitors, the t-test is used in our experiment. The analysis data is the classification results of 30 runs independently obtained by each algorithm on 15 datasets. The experiments were conducted using a two-tailed t-test with a significance level of 5%. Table 9 shows the analysis results between LAPSO-DR and these eight algorithms on 18 datasets. In general, "+ " and "−" indicate that LAPSO-DR is significantly better or worse than competitors, and "=" indicates that there is no significant difference between them.

From Tables 3 and 9, we can see that the classification accuracy obtained by LAPSO-DR has no significant difference compared to those of GA, PSO, BBPSO, CSO, and HPSO-SSM on the Segmentation dataset. At the same time, it is significantly better than BGWOPSO, PPSO, and PSO-EMT. On the WDBC dataset, there is no significant difference between LAPSO-DR and competitors. On the Ionosphere, Sonar, Hill Valley, Musk1, Divorce, Arrhythmia, LSVT, Madelon, Isolet5, MFS, Leukemia1, Leukemia2, and Leukemia3 datasets, LAPSO-DR performs significantly better than the other eight algorithms. Furthermore, for ORL32 dataset, LAPSO-DR performs significantly better than BBPSO, HPSO-SSM, and PSO-EMT, but there is no significant difference compared to the other five algorithms. On the SRBCT and DLBCL datasets, there is no significant difference between LAPSO-DR and PSO-EMT, but the former is significantly better than the other seven competitors.

Based on the statistical analysis of experimental results obtained from the independent t-test, LAPSO-DR algorithm shows the superiority of performance for classification datasets, comparing to eight state-of-the-art wrapper-based algorithms (GA, PSO, BBPSO, CSO, HPSO-SSM, BGWOPSO, PPSO, and PSO-EMT). Furthermore, the experimental results shown in the previous section, which demonstrates the performance of LAPSO-DR significantly better than those competitors, has also been validated effectively.

4.4.3. Analyses on main strategies

The proposed LAPSO-DR algorithm has three main strategies, which are the hybrid initialization strategy, the leader-adaptive strategy and the dimensionality reduction strategy based on MB. To test the effectiveness of the three strategies, we conduct deduplication experiments on these 18 datasets. Firstly, to verify the effect of the hybrid initialization strategy, we set up variant I of LAPSO-DR, called LAPSO-DR-A, which adopts the ordinary random initialization strategy in initialization phase. Secondly, to verify the effect of the leader-adaptive strategy, we design variant II (LAPSO-DR-B) of LAPSO-DR, which has only one social leader $Gbest$. Finally, variant III (LAPSO-DR-C) of LAPSO-DR is set to validate the effect of the dimensionality reduction strategy base on MB. There is no dimensionality reduction strategy in LAPSO-DR-C.

In particular, these three variants have the same parameters settings as LAPSO-DR. Each variant and the LAPSO-DR are run independently 30 times on 18 datasets, and their average classification results are used for comparison. Table 10 lists the experimental results obtained by LAPSO-DR and its three variants on 18 datasets, where "d" represents the size of selected feature subset and "acc" represents the average classification accuracy (%).

(1) LAPSO-DR VS LAPSO-DR-A. Overall, the average classification accuracies obtained by LAPSO-DR are higher than those of LAPSO-DR-A. In particular, for low-dimensional datasets, there are no significant difference between LAPSO-DR and LAPSO-DR-A in the classification accuracy on Segmentation and WDBC datasets. However, the classification accuracies obtained by LAPSO-DR are much better than those of LAPSO-DR-A on Ionosphere and Sonar datasets. Furthermore, the classification accuracies obtained by LAPSO-DR are much better than those of LAPSO-DR-A on medium-dimensional and high-dimensional datasets. Based on the experimental results listed in Table 10, it can be seen that the hybrid initialization strategy has a certain positive impact on improving the classification accuracy of LAPSO-DR.

When comparing the sizes of selected feature subsets, there is little difference between LAPSO-DR and LAPSO-DR-A on low-dimensional and medium-dimensional datasets. However, LAPSO-DR selects fewer features on high-dimensional datasets when compared with LAPSO-DR-A. Especially, the sizes of selected feature subsets obtained by LAPSO-DR are significantly smaller than that of LAPSO-DR-A on high-dimensional datasets. This indicates that the hybrid initialization strategy can avoid some redundant features from being selected at the initial stage.

Table 9
Results of *t*-test between LAPSO-DR and eight algorithms in terms of classification accuracy (by KNN).

Dataset	GA	PSO	BBPSO	CSO	BGWOPSO	HPSO-SSM	PPSO	PSO-EMT
Segmentation	=	=	=	=	+	=	+	+
WDBC	=	=	=	=	=	=	=	=
Ionosphere	+	+	+	+	+	+	+	+
Sonar	+	+	+	+	+	+	+	+
Hill-Valley	+	+	+	+	+	+	+	+
Musk1	+	+	+	+	+	+	+	+
Divorce	+	+	+	+	+	+	+	+
Arrhythmia	+	+	+	+	+	+	+	+
LSVT	+	+	+	+	+	+	+	+
Madelon	+	+	+	+	+	+	+	+
Isolet5	+	+	+	+	+	+	+	+
MFS	+	+	+	+	+	+	+	+
ORL32	=	=	+	=	=	+	=	+
SRBCT	+	+	+	+	+	+	+	=
DLBCL	+	+	+	+	+	+	+	=
Leukemia1	+	+	+	+	+	+	+	+
Leukemia2	+	+	+	+	+	+	+	+
Leukemia3	+	+	+	+	+	+	+	+

Table 10
Experimental results of LAPSO-DR in the four cases: without hybrid initialization(LAPSO-DR-A), without the leader-adaptive strategy (LAPSO-DR-B), without the dimensionality reduction strategy (LAPSO-DR-C) and the full LAPSO-DR (LAPSO-DR).

Dataset	LAPSO-DR-A		LAPSO-DR-B		LAPSO-DR-C		LAPSO-DR	
	d	*acc*(%)	*d*	*acc*(%)	*d*	*acc*(%)	*d*	*acc*(%)
Segmentation	3.10	96.68(=)	3.23	96.58(=)	**3.06**	96.67(=)	**3.06**	**96.76**
WDBC	5.23	94.62(=)	4.73	94.78(=)	4.57	94.91(=)	**4.53**	**94.97**
Ionosphere	6.57	88.92(+)	6.00	89.75(+)	**5.33**	88.10(+)	7.93	**93.14**
Sonar	19.73	84.89(+)	**16.07**	83.23(+)	17.50	83.39(+)	27.06	**93.28**
Hill-Valley	28.63	53.98(+)	11.60	52.25(+)	**11.43**	53.38(+)	28.97	**60.26**
Musk1	54.93	88.25(+)	51.03	86.13(+)	**49.16**	87.58(+)	71.10	**92.63**
Divorce	3.17	97.06(+)	2.97	96.80(+)	**2.50**	96.99(+)	4.57	**98.63**
Arrhythmia	96.53	57.99(+)	68.46	57.23(+)	**64.23**	58.96(+)	91.93	**63.53**
LSVT	127.03	79.47(+)	**47.40**	78.86(+)	51.27	81.23(+)	96.93	**90.96**
Madelon	155.40	61.15(+)	**9.07**	86.30(+)	11.27	88.66(=)	9.73	**88.97**
Isolet5	200.70	86.60(+)	194.90	85.53(+)	226.80	86.18(+)	**183.37**	**88.53**
MFS	153.73	97.62(+)	40.83	98.18(=)	53.50	98.23(=)	**39.97**	**98.24**
ORL32	434.60	89.94(=)	283.90	90.36(=)	259.47	90.08(=)	**221.9**	**91.44**
SRBCT	891.80	85.20(+)	**176.30**	84.67(+)	187.83	88.93(+)	182.93	**95.33**
DLBCL	2114.00	83.77(+)	297.67	88.12(+)	354.20	88.12(+)	**278.03**	**94.35**
Leukemia1	1904.50	85.45(+)	111.60	90.45(+)	125.00	90.45(+)	**99.50**	**93.18**
Leukemia2	2502.00	86.82(+)	97.80	96.36(+)	118.00	97.73(+)	**88.60**	**98.64**
Leukemia3	4279.90	85.45(+)	**305.80**	91.82(+)	374.70	88.18(+)	408.50	**92.73**

Further analysis shows that the hybrid initialization strategy of LAPSO-DR can effectively utilize the correlation information between features and labels, which can eliminate some of the redundant features at the initial stage. At the same time, some particles in the hybrid initialization stage also adopt the common initialization strategy, which is beneficial to increase the diversity of the population. Therefore, the experimental results show that LAPSO-DR is better than LAPSO-DR-A in terms of classification accuracy and the size of selected feature subset, which proves the effectiveness of the proposed hybrid initialization strategy.

(2) LAPSO-DR VS LAPSO-DR-B. From Table 10, it can be seen that the classification accuracies obtained by LAPSO-DR are much better than that of LAPSO-DR-B on 15 out of 18 datasets. In contrast, the classification accuracies obtained by LAPSO-DR and LAPSO-DR-B are not much different on three datasets (WDBC, MFS, and ORL32). Based on this, it can be found that the leader-adaptive strategy is beneficial to improve the classification accuracy of LAPSO-DR.

Considering of the sizes of selected feature subsets, LAPSO-DR is slightly smaller than LAPSO-DR-B on high-dimensional datasets, while the later is smaller than the former on low-dimensional and medium-dimensional datasets. In other words, these experimental results indicate that the leader-adaptive strategy does not have a good effect on LAPSO-DR in terms of removing redundant (or irrelevant) features.

Based on experimental results, this is because the leader-adaptive strategy can help to jump out of the local optimum solutions by replacing the learning exemplars of particles in long-term stagnation, so that performance of LAPSO-DR can be further improved. For the sizes of selected feature subsets, there is generally not much difference between LAPSO-DR and LAPSO-DR, which indicates that the leader-adaptive strategy does eliminate redundant features while improving the classification accuracy. By analyzing the classification accuracy and the sizes of feature subsets obtained by LAPSO-DR, we can see that the leader-adaptive strategy has a positive effect on the performance of LAPSO-DR.

(3) LAPSO-DR VS LAPSO-DR-C. It can be seen from Table 10 that the average classification accuracies obtained by LAPSO-DR are better than those of LAPSO-DR-C. Specifically, the classification accuracies obtained by LAPSO-DR are not significantly different from LAPSO-DR-C on five datasets (Segmentation, WDBC, Madelon, MFS, and ORL32). However, the classification accuracies obtained by LAPSO-DR are much better than LAPSO-DR-C on 12 out of 18 datasets. In general, it is found that the dimensionality reduction strategy based on MB is beneficial to the classification accuracy of LAPSO-DR.

In terms of the sizes of selected feature subsets, LAPSO-DR performs slightly worse than LAPSO-DR-C on low-dimensional and medium-dimensional datasets, but the former selects fewer features than the later on high-dimensional datasets. Through further analysis, it can be seen that the dimensionality reduction strategy based on MB can effectively eliminate redundant (or irrelevant) features on high-dimensional datasets with large density.

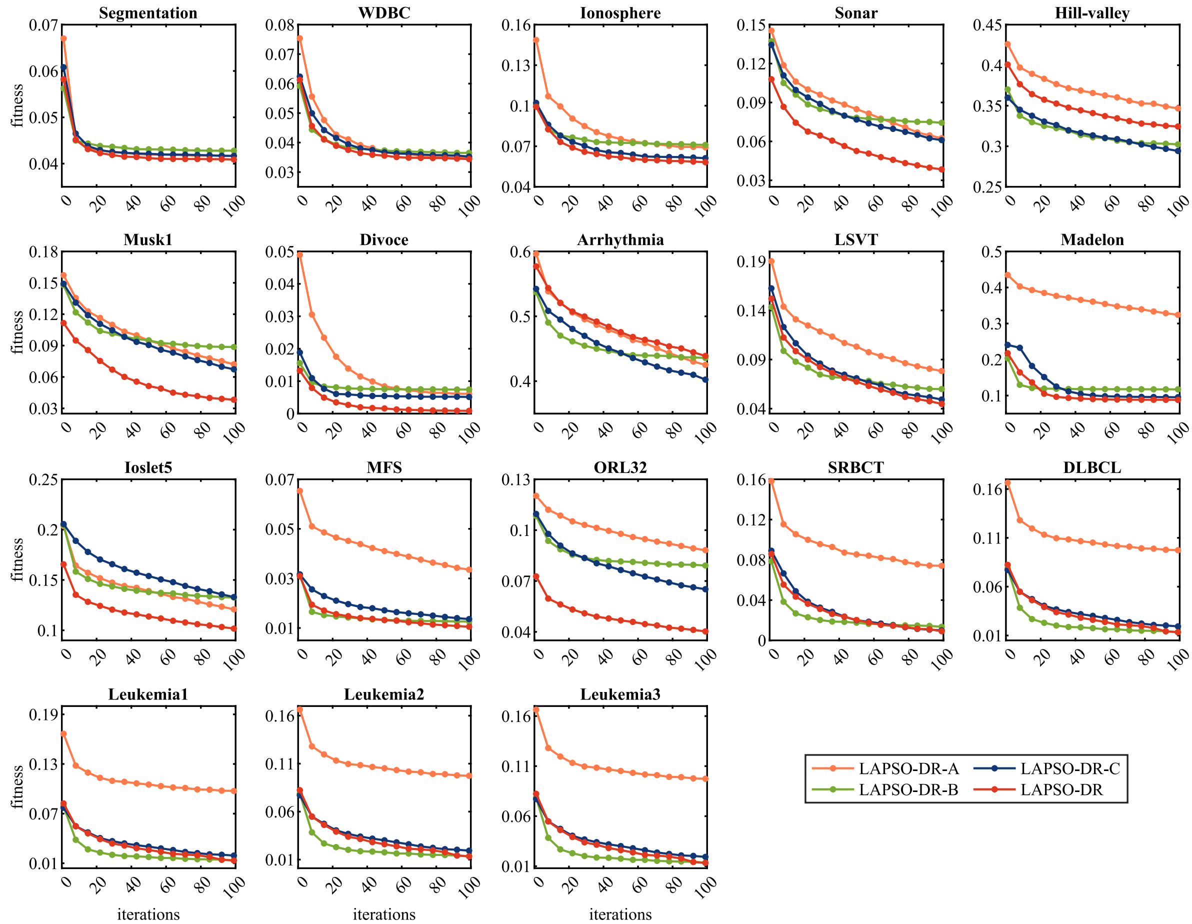

Fig. 5. The fitness curves obtained by LAPSO-DR and the three variants.

Through experimental results, it is found that the dimensionality reduction strategy is beneficial to the classification accuracy of LAPSO-DR and has a good effect on removing the redundant features of high-dimensional datasets. According to the dimensionality reduction strategy, the elite sub-swarm can help to further improve the quality of elite individuals, so as to improve the overall effect of the population. In the analysis of the sizes of selected feature subsets, for high-dimensional datasets with large density, the dimensionality reduction strategy can effectively eliminate redundant features and improve the classification accuracy achieved by LAPSO-DR. However, for low-dimensional and medium-dimensional datasets with low density, the dimension reduction effect of this strategy is not obvious. Therefore, the dimensionality reduction strategy is beneficial to improve the classification accuracy, and has a better dimension reduction effect on high-dimensional datasets.

To further analyze the performance of the proposed LAPSO-DR, the fitness curves of four algorithm (LAPSO-DR, LAPSO-DR-A, LAPSO-DR-B, and LAPSO-DR-C) on 18 datasets are shown in Fig. 5. It can be seen that the fitnesses of LAPSO-DR are smaller than that of three variants on 16 out of 18 datasets, indicating that the overall performance of LAPSO-DR is better than that of its variants. In addition, the elimination ratios of features achieved by LAPSO-DR and the three variants are shown in Fig. 6, where the best result is marked by the pentagram. From Fig. 6, it can be seen that LAPSO-DR has a better effect than LAPSO-DR-A in terms of dimensionality reduction on 12 out of 18 datasets. At the same time, LAPSO-DR and LAPSO-DR-B have better effect than LAPSO-DR-C on high-dimensional datasets. Through further analysis, this phenomenon indicates that the dimensionality reduction strategy based on MB plays an important role in reducing the size of the optimal feature subset.

4.4.4. Parameter sensitivity analysis

The division ratio ($DivRatio$), which is employed to divide the population into two parts at initial stage, is an important parameter for the hybrid initialization strategy in LAPSO-DR. The value of $DivRatio$ plays a crucial role in the performance of LAPSO-DR. From our analysis, more particles have different initialization ranges when the value of $DivRatio$ is too large. In this way, it is difficult for some related features to be selected, resulting in the poor classification accuracy of LAPSO-DR. In comparison, more particles are randomly initialized directly within the whole search space when the value of $DivRatio$ is too small. In this event, the population may easily lose diversity. Therefore, an experimental study about $DivRatio$ was conducted in our work, and the value of $DivRatio$ was set to be $\{9:1, 8:2, 7:3, 6:4, 5:5\}$ for finding the appropriate value. The experimental results obtained by LAPSO-DR with different values on 15 datasets are shown in Table 11, where "d" and "acc" represent the average size of selected features and the average classification accuracy, respectively.

From Table 11, it can be found that the LAPSO-DR selects fewer features when the value of $DivRatio$ increases, and this phenomenon is particularly obvious on high-dimensional datasets. When $DivRatio = 9:1$, LAPSO-DR achieves the minimum sizes of selected feature subsets on more than half of all these datasets. In particular, for five datasets (Segmentation, WDBC, Arrhythmia, LSVT, and MFS), LAPSO-DR achieves the smallest sizes of selected feature subsets and the highest classification accuracies under $DivRatio = 9:1$. When $DivRatio = 8:2$, LAPSO-DR has the best classification accuracy on the ORL32 dataset, and it achieves the least number of selected features on the DLBCL dataset. Furthermore, LAPSO-DR achieves the best classification accuracies and smaller sizes of selected feature subsets on 10 out of 18 datasets when $DivRatio = 7:3$. When $DivRatio =$

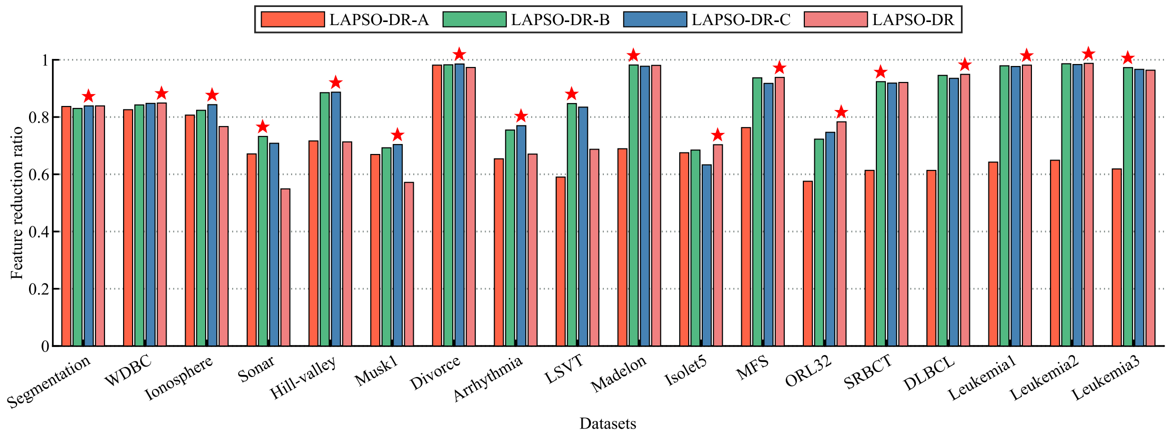

Fig. 6. The elimination ratios of features obtained by LAPSO-DR and three variants.

Table 11
The experimental results obtained by LAPSO-DR with different values of $DivRatio$.

Dataset	$DivRatio = 9:1$		$DivRatio = 8:2$		$DivRatio = 7:3$		$DivRatio = 6:4$		$DivRatio = 5:5$	
	d	$acc(\%)$	d	$acc(\%)$	d	$acc(\%)$	d	$acc(\%)$	d	$acc(\%)$
Segmentation	**3.00**	**96.83**	7.00	96.65	3.06	96.76	3.37	96.38	3.33	**96.83**
WDBC	**3.93**	**96.14**	10.77	95.54	4.53	94.97	4.96	94.33	4.57	94.46
Ionosphere	5.4	93.05	6.6	88.89	7.93	**93.14**	8.33	89.49	**5.17**	88.38
Sonar	**15.47**	89.52	23.70	84.09	27.06	**93.28**	15.93	81.94	16.93	91.51
Hill-Valley	12.4	59.61	21.67	52.03	28.97	**60.26**	**10.50**	54.16	10.70	52.29
Musk1	**42.70**	92.40	71.07	88.76	71.10	**92.63**	52.80	85.62	75.16	88.09
Divorce	3.10	98.50	2.97	97.25	4.57	**98.63**	**2.83**	96.67	2.93	96.41
Arrhythmia	**58.77**	**64.90**	77.40	57.60	91.93	63.53	68.93	57.67	71.33	56.81
LSVT	**50.60**	**91.40**	89.07	81.23	96.93	90.96	101.57	78.86	98.63	80.09
Madelon	9.70	88.93	10.13	88.45	9.73	**88.97**	**9.47**	85.76	12.03	87.77
Isolet5	187.43	88.14	186.97	87.97	**183.37**	**88.53**	186.37	85.09	248.60	86.20
MFS	**38.60**	**98.95**	45.23	98.33	39.97	98.24	41.23	98.19	162.20	97.77
ORL32	**217.37**	92.53	218.50	**93.31**	221.9	91.44	291.77	89.50	475.67	89.94
SRBCT	**147.90**	94.13	227.90	88.53	182.93	**95.33**	232.97	85.47	219.37	86.00
DLBCL	247.77	93.77	**231.30**	86.09	278.03	**94.35**	428.70	89.86	467.10	87.83
Leukemia1	**71.60**	91.36	73.80	89.55	99.50	93.18	99.00	**94.55**	92.40	90.00
Leukemia2	**52.30**	96.82	57.50	97.27	88.60	**98.64**	233.70	87.73	93.30	97.27
Leukemia3	235.00	**93.18**	**155.80**	91.82	408.50	92.73	199.10	90.45	299.50	**93.18**

6 : 4 ($DivRatio = 5 : 5$), LAPSO-DR obtains the highest classification accuracies only on Segmentation and Leukemia1 datasets, and obtains the minimum sizes of selected feature subsets on 4 out of 18 datasets. Therefore, we consider $DivRatio = 7 : 3$ as the optimal setting since the classification accuracy is more weighted in dataset classification problem.

Furthermore, the fitness curves achieved by LAPSO-DR with different values of $DivRatio$ are shown in Fig. 7. It can be seen that the fitnesses obtained by LAPSO-DR with $DivRatio = 7 : 3$ is smaller on most of these 18 datasets, indicating that LAPSO-DR has the better performance when $DivRatio = 7 : 3$. Fig. 8 shows the feature elimination ratios of LAPSO-DR with different values of $DivRatio$. Obviously, it can be seen from Fig. 8 that LAPSO-DR has the best performance in terms of eliminating redundant (irrelevant) features when $DivRatio = 9 : 1$. And in most cases, the larger the value of $DivRatio$, the better the dimensionality reduction effect of LAPSO-DR.

5. Conclusion

In order to better solve FS task in datasets classification problem, this paper proposes a leader-adaptive particle swarm optimization algorithm with dimensionality reduction strategy (LAPSO-DR). Firstly, the features are rearranged based on their SU values, and the population is divided into two parts with different initialization ranges. This hybrid initialization strategy not only eliminates some redundant features at initial stage, but also forms a diverse population. And then, a leader-adaptive strategy is adopted to provide learning exemplars with high quality for these stagnant individuals, so as to enhance the exploitation ability of the population. At the same time, a new inter-particle learning strategy is introduced, in which different dimensions of each particle can learn from different particles, to improve the exploration ability of the population. Finally, the dimensionality reduction strategy based on MB is introduced to reduce the size of the optimal feature subset. The proposed LAPSO-DR algorithm is compared with eight representative algorithms on 18 datasets. The experimental results show that the LAPSO-DR has higher classification performance on 17 out of 18 datasets, which indicates that LAPSO-DR has a good effect on solving the FS problem. In addition, the effects of the hybrid initialization strategy, the leader-adaptive strategy, and the dimensionality reduction strategy on LAPSO-DR are also verified. Although LAPSO-DR achieves the better results in terms of the classification accuracy, compared with 8 state-of-art wrapper-based FS algorithms, it is slightly inferior than competitors in dimensionality reduction. Therefore, we will continue to investigate how to improve the performance of LAPSO-DR in terms of reducing the size of the optimal feature subset.

Our future work will be devoted to two aspects: (1) considering FS problems such as semi-supervised nd adopting LAPSO-DR for application; (2) considering how LAPSO-DR would be improved to apply multi-objective high-dimensional feature selection.

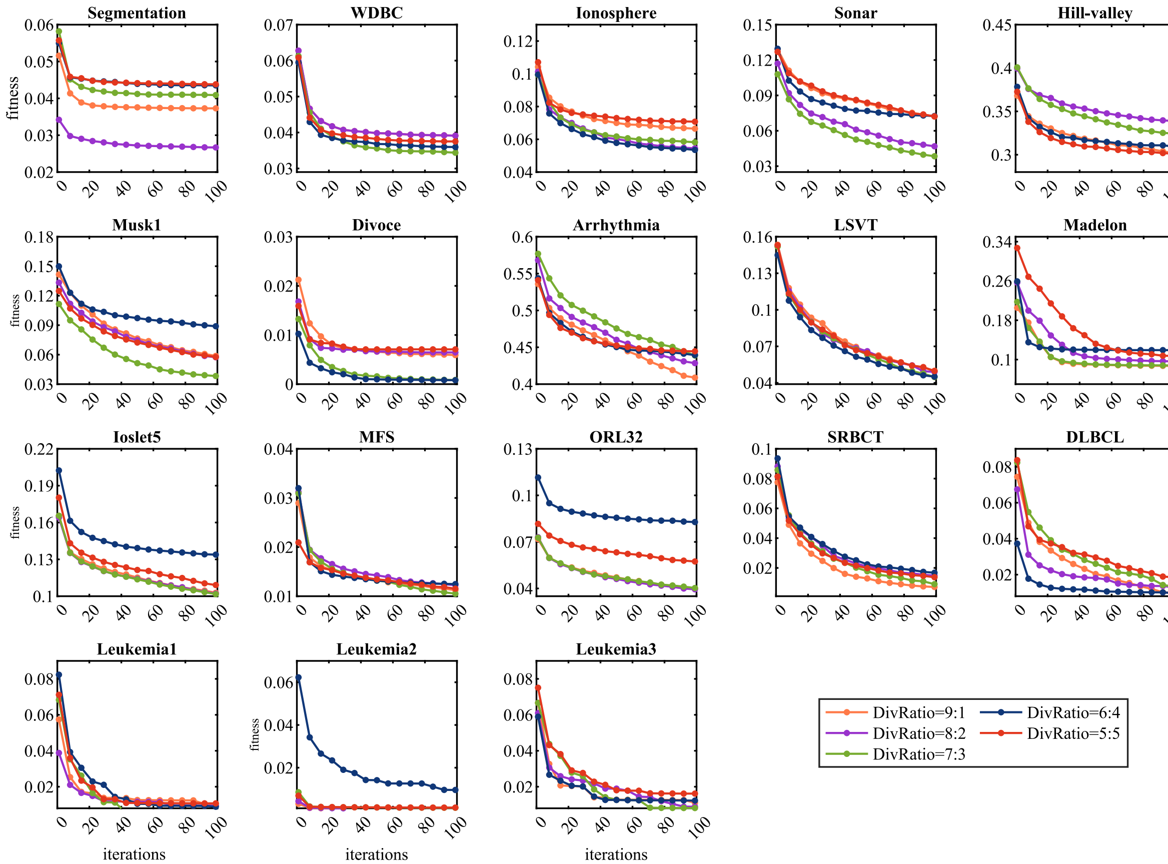

Fig. 7. The fitness curves obtained by LAPSO-DR with different values of *DivRatio*.

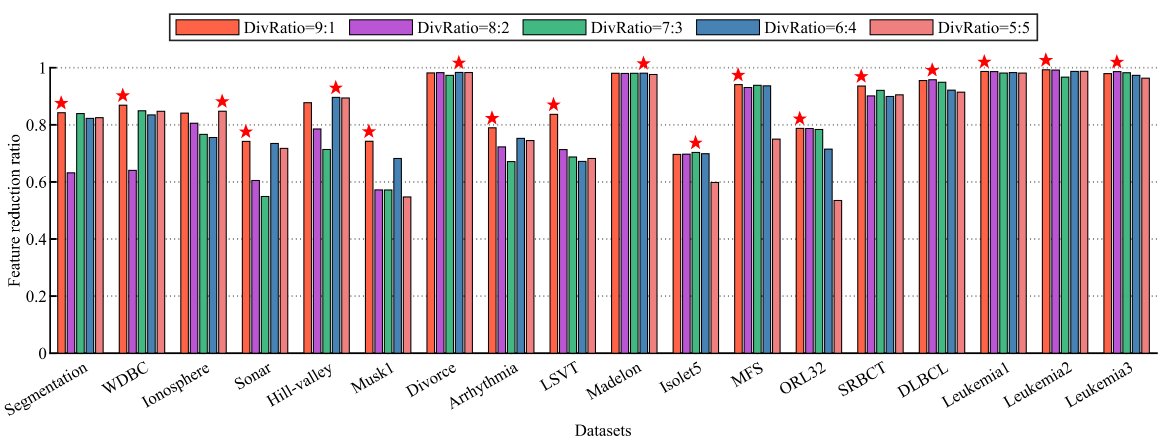

Fig. 8. The elimination ratios of features obtained by LAPSO-DR with different values of *DivRatio*.

CRediT authorship contribution statement

Shanshan Yang: Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Bo Wei:** Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Li Deng:** Visualization, Software, Investigation, Data curation. **Xiao Jin:** Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Mingfeng Jiang:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Formal analysis. **Yanrong Huang:** Visualization, Software, Investigation, Data curation. **Feng Wang:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Bo Wei reports financial support was provided by the Basic Public Welfare Research Project of Zhejiang Province. Bo Wei reports financial support was provided by the National Natural Science Foundation of China. Yanrong Huang reports financial support was provided by the National Natural Science Foundation of China. Mingfeng Jiang reports financial support was provided by the Key Research and Development Program of Zhejiang Province. Mingfeng Jiang reports financial support

was provided by Joint Found of Zhejiang Provincial Natural Science Foundation. Bo Wei reports financial support was provided by the Research Fund Project of Zhejiang Sci-Tech University Longgang Research Institute. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This study is funded by the Basic Public Welfare Research Project of Zhejiang Province (LGF22F020020), the National Natural Science Foundation of China (61806204, 72101235, 62011530130, 62272415), Scientific Research Starting Foundation of Zhejiang Sci-Tech University (20032309-Y), the Key Research and Development Program of Zhejiang Province (2020C03060), Joint Fund of Zhejiang Provincial Natural Science Foundation (LSZ19F010001), and the Research Fund Project of Zhejiang Sci-Tech University Longgang Research Institute (LGYJY2023003).

References

[1] Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, M. Barni, Secure detection of image manipulation by means of random feature selection, IEEE Trans. Inf. Forensics Secur. 14 (9) (2019) 2454–2469, http://dx.doi.org/10.1109/TIFS.2019.2901826.
[2] X. Wang, F. Wang, Q. He, Y. Guo, A multi-swarm optimizer with a reinforcement learning mechanism for large-scale optimization, Swarm Evol. Comput. 86 (2024) 101486, http://dx.doi.org/10.1016/j.swevo.2024.101486.
[3] E.J. Keogh, A. Mueen, Curse of dimensionality, in: C. Sammut, G.I. Webb (Eds.), Encyclopedia of Machine Learning and Data Mining, Springer, 2017, pp. 314–315, http://dx.doi.org/10.1007/978-1-4899-7687-1_192.
[4] J.D. Li, K.W. Cheng, S.H. Wang, F. Morstatter, R.P. Trevino, J.L. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (6) (2017) 1–45, http://dx.doi.org/10.1145/3136625.
[5] H.Y. Liu, M.C. Zhou, Q. Liu, An embedded feature selection method for imbalanced data classification, IEEE-CAA J. Autom. Sin. 6 (3) (2019) 703–715, http://dx.doi.org/10.1109/JAS.2019.1911447.
[6] S. Huang, Z. Wang, Y. Ge, F. Wang, A coevolutionary estimation of distribution algorithm based on dynamic differential grouping for mixed-variable optimization problems, Expert Syst. Appl. 245 (2024) 123122, http://dx.doi.org/10.1016/j.eswa.2023.123122.
[7] S. Li, F. Wang, Q. He, X. Wang, Deep reinforcement learning for multi-objective combinatorial optimization: A case study on multi-objective traveling salesman problem, Swarm Evol. Comput. 83 (2023) 101398, http://dx.doi.org/10.1016/j.swevo.2023.101398.
[8] B.H. Nguyen, B. Xue, M.J. Zhang, A survey on swarm intelligence approaches to feature selection in data mining, Swarm Evol. Comput. 54 (2020) 100663, http://dx.doi.org/10.1016/j.swevo.2020.100663.
[9] O. Tarkhaneh, T.T. Nguyen, S. Mazaheri, A novel wrapper-based feature subset selection method using modified binary differential evolution algorithm, Inform. Sci. 565 (2021) 278–305, http://dx.doi.org/10.1016/J.INS.2021.02.061.
[10] S. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, Expert Syst. Appl. 41 (4) (2014) 2052–2064, http://dx.doi.org/10.1016/j.eswa.2013.09.004.
[11] Y.C. Wan, M.W. Wang, Z.W. Ye, X.D. Lai, A feature selection method based on modified binary coded ant colony optimization algorithm, Appl. Soft Comput. 49 (2016) 248–258, http://dx.doi.org/10.1016/j.asoc.2016.08.011.
[12] N. Kushwaha, M. Pant, Link based bpso for feature selection in big data text clustering, Future Gener. Comput. Syst. 82 (2018) 190–199, http://dx.doi.org/10.1016/j.future.2017.12.005.
[13] Y. Hu, Y. Zhang, D. Gong, Multiobjective particle swarm optimization for feature selection with fuzzy cost, IEEE Trans. Cybern. 51 (2) (2021) 874–888, http://dx.doi.org/10.1109/TCYB.2020.3015756.
[14] H. Bayati, M.B. Dowlatshahi, M. Paniri, MLPSO: A filter multi-label feature selection based on particle swarm optimization, in: 2020 25th International Computer Conference, Computer Society of Iran, CSICC, 2020, pp. 1–6, http://dx.doi.org/10.1109/CSICC49403.2020.9050087.
[15] X.F. Song, Y. Zhang, D.W. Gong, X.Y. Sun, Feature selection using bare-bones particle swarm optimization with mutual information, Pattern Recognit. 112 (2021) 107804, http://dx.doi.org/10.1016/j.patcog.2020.107804.
[16] K. Chen, B. Xue, M. Zhang, F. Zhou, Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization, IEEE Trans. Evol. Comput. 26 (3) (2022) 446–460, http://dx.doi.org/10.1109/TEVC.2021.3100056.
[17] B. Tran, B. Xue, M. Zhang, A new representation in pso for discretization-based feature selection, IEEE Trans. Cybern. 48 (6) (2018) 1733–1746, http://dx.doi.org/10.1109/TCYB.2017.2714145.
[18] B. Tran, B. Xue, M.J. Zhang, Variable-length particle swarm optimization for feature selection on high-dimensional classification, IEEE Trans. Evol. Comput. 23 (3) (2018) 473–487, http://dx.doi.org/10.1109/TEVC.2018.2869405.
[19] L.T. Qu, W.B. He, J.F. Li, H. Zhang, C. Yang, B. Xie, Explicit and size-adaptive pso-based feature selection for classification, Swarm Evol. Comput. 77 (2023) 101249, http://dx.doi.org/10.1016/j.swevo.2023.101249.
[20] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995, pp. 39–43, http://dx.doi.org/10.1109/MHS.1995.494215.
[21] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of ICNN'95-International Conference on Neural Networks,Vol. 4, IEEE, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.
[22] Y. Xue, T. Tang, W. Pang, X.A. Liu, Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers, Appl. Soft Comput. 88 (2020) 106031, http://dx.doi.org/10.1016/j.asoc.2019.106031.
[23] X. Song, Y. Zhang, D. Gong, H. Liu, W. Zhang, Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data, IEEE Trans. Evol. Comput. 27 (3) (2023) 595–609, http://dx.doi.org/10.1109/TEVC.2022.3175226.
[24] A. Rashno, M. Shafipour, S. Fadaei, Particle ranking: An efficient method for multi-objective particle swarm optimization feature selection, Knowl.-Based Syst. 245 (2022) 108640, http://dx.doi.org/10.1016/j.knosys.2022.108640.
[25] J.R. Gao, Z.Q. Wang, T. Jin, J.J. Cheng, Z.Y. Lei, S.C. Gao, Information gain ratio-based subfeature grouping empowers particle swarm optimization for feature selection, Knowl.-Based Syst. 286 (2024) 111380, http://dx.doi.org/10.1016/j.knosys.2024.111380.
[26] H. Djellali, N. Ghoualmi, Improved chaotic initialization of particle swarm applied to feature selection, in: 2019 International Conference on Networking and Advanced Systems, ICNAS, 2019, pp. 1–5, http://dx.doi.org/10.1109/ICNAS.2019.8807837.
[27] Y. Xue, W. Jia, A.X. Liu, A particle swarm optimization with filter-based population initialization for feature selection, in: 2019 IEEE Congress on Evolutionary Computation, CEC, 2019, pp. 1572–1579, http://dx.doi.org/10.1109/CEC.2019.8790156.
[28] B.H. Nguyen, B. Xue, M. Zhang, A constrained competitive swarm optimiser with an svm-based surrogate model for feature selection, IEEE Trans. Evol. Comput. 28 (1) (2022) 2–16, http://dx.doi.org/10.1109/TEVC.2022.3197427.
[29] K. Chen, F.Y. Zhou, X.F. Yuan, Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection, Expert Syst. Appl. 128 (2019) 140–156, http://dx.doi.org/10.1016/j.eswa.2019.03.039.
[30] H.B. Nguyen, B. Xue, P. Andreae, Surrogate-model based particle swarm optimisation with local search for feature selection in classification, in: European Conference on Applications of Evolutionary Computation, 2017, pp. 487–505, http://dx.doi.org/10.1007/978-3-319-55849-3_32.
[31] D. Juhini, B.H. Nguyen, B. Xue, Multi-label feature selection using particle swarm optimization: Novel local search mechanisms, in: 2019 IEEE Symposium Series on Computational Intelligence, SSCI, 2019, pp. 1762–1769, http://dx.doi.org/10.1109/SSCI44817.2019.9002734.
[32] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, Appl. Soft Comput. 43 (2016) 117–130, http://dx.doi.org/10.1016/j.asoc.2016.01.044.
[33] M. Paniri, M.B. Dowlatshahi, H. Nezamabadi-pour, Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection, Swarm Evol. Comput. 64 (2021) 100892, http://dx.doi.org/10.1016/j.swevo.2021.100892.
[34] M.A. Awadallah, A.I. Hammouri, M.A. Al-Betar, M.S. Braik, M.A. Elaziz, Binary horse herd optimization algorithm with crossover operators for feature selection, Comput. Biol. Med. 141 (2022) 105152, http://dx.doi.org/10.1016/j.compbiomed.2021.105152.
[35] F. Karimi, M.B. Dowlatshahi, A. Hashemi, SemiACO: A semi-supervised feature selection based on ant colony optimization, Expert. Syst. Appl. 214 (2023) 119130, http://dx.doi.org/10.1016/j.eswa.2022.119130.
[36] M. Paniri, M.B. Dowlatshahi, H. Nezamabadi-pour, MLACO: A multi-label feature selection algorithm based on ant colony optimization, Knowl.-Based Syst. 192 (2020) 105285, http://dx.doi.org/10.1016/j.knosys.2019.105285.
[37] A. Hashemi, M. Joodaki, N.Z. Joodaki, M.B. Dowlatshahi, Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: A case study in ensemble feature selection, Appl. Soft Comput. 124 (2022) 109046, http://dx.doi.org/10.1016/j.asoc.2022.109046.
[38] Y. Zhou, W. Zhang, J. Kang, X. Zhang, X. Wang, A problem-specific non-dominated sorting genetic algorithm for supervised feature selection, Inform. Sci. 547 (2021) 841–859, http://dx.doi.org/10.1016/j.ins.2020.08.083.

[39] M. Braik, M.A. Awadallah, M.A. Al-Betar, A.I. Hammouri, O.A. Alzubi, Cognitively enhanced versions of capuchin search algorithm for feature selection in medical diagnosis: a covid-19 case study, Cogn. Comput. 15 (2023) 1884–1921, http://dx.doi.org/10.1007/s12559-023-10149-0.

[40] Y. Zhao, J. Dong, X. Li, H. Chen, S. Li, A binary dandelion algorithm using seeding and chaos population strategies for feature selection, Appl. Soft Comput. 125 (2022) 109166, http://dx.doi.org/10.1016/j.asoc.2022.109166.

[41] M.A. Awadallah, M.A.A. Betar, M.S. Braik, A.I. Hammouri, I.A. Doush, R.A. Zitar, An enhanced binary rat swarm optimizer based on local-best concepts of pso and collaborative crossover operators for feature selection, Comput. Biol. Med. 147 (2022) 105675, http://dx.doi.org/10.1016/j.compbiomed.2022.105675.

[42] M.A. Awadallah, M.S. Braik, M.A. Al-Betar, I. Abu Doush, An enhanced binary artificial rabbits optimization for feature selection in medical diagnosis, Neural Comput. Appl. 35 (27) (2023) 20013–20068, http://dx.doi.org/10.1007/s00521-023-08812-6.

[43] Z. Hu, X. Yu, Reinforcement learning-based comprehensive learning grey wolf optimizer for feature selection, Appl. Soft Comput. 149 (2023) 110959, http://dx.doi.org/10.1016/j.asoc.2023.110959.

[44] D. Koller, M. Sahami, Toward optimal feature selection, in: Morgan Kaufmann Ser. Data Manage. Syst, 1996.

[45] J. Lee, J.Y. Jeong, J.C. Hyuck, Markov blanket-based universal feature selection for classification and regression of mixed-type data, Expert Syst. Appl. 158 (2020) 113398, http://dx.doi.org/10.1016/J.ESWA.2020.113398.

[46] Y.F. Zeng, J. Luo, S.Y. Lin, Classification using markov blanket for feature selection, in: The 2009 IEEE International Conference on Granular Computing, IEEE Computer Society, 2009, pp. 743–747, http://dx.doi.org/10.1109/GRC.2009.5255023.

[47] L. Yu, H. Liu, Redundancy based feature selection for microarray data, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 737–742, http://dx.doi.org/10.1145/1014052.1014149.

[48] Z. x. Zhu, Y.S. Ong, M. Dash, Markov blanket-embedded genetic algorithm for gene selection, Pattern Recognit. 40 (11) (2007) 3236–3248, http://dx.doi.org/10.1016/j.patcog.2007.02.007.

[49] Z.S. Hua, J. Zhou, Y. Hua, W. Zhang, Strong approximate markov blanket and its application on filter-based feature selection, Appl. Soft Comput. 87 (2020) 105957, http://dx.doi.org/10.1016/j.asoc.2019.105957.

[50] J. Zhou, Q. Wu, M. Zhou, J. Wen, Y. Al-Turki, A. Abusorrah, Lagam: A length-adaptive genetic algorithm with markov blanket for high-dimensional feature selection in classification, IEEE Trans. Cybern. 53 (11) (2023) 6858–6869, http://dx.doi.org/10.1109/TCYB.2022.3163577.

[51] S.A. Teukolsky, B.P. Flannery, W. Press, W. Vetterling, Numerical recipes in c, SMR 693 (1) (1992) 59–70.

[52] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: T. Fawcett, N. Mishra (Eds.), Machine Learning, Proceedings of the Twentieth International Conference, AAAI Press, 2003, pp. 856–863.

[53] Q.B. Song, J.J. Ni, G.T. Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, IEEE Trans. Knowl. Data Eng. 25 (1) (2011) 1–14, http://dx.doi.org/10.1109/TKDE.2011.181.

[54] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295, http://dx.doi.org/10.1109/TEVC.2005.857610.

[55] K. Chen, B. Xue, M.J. Zhang, F.Y. Zhou, An evolutionary multitasking-based feature selection method for high-dimensional classification, IEEE Trans. Cybern. 52 (7) (2020) 7172–7186, http://dx.doi.org/10.1109/TCYB.2020.3042243.

[56] T.Y. Li, J.Y. Shi, W. Deng, Z.D. Hu, Pyramid particle swarm optimization with novel strategies of competition and cooperation, Appl. Soft Comput. 121 (2022) 108731, http://dx.doi.org/10.1016/j.asoc.2022.108731.

[57] Q. Al-Tashi, S.J.A. Kadir, H.M. Rais, S. Mirjalili, H. Alhussian, Binary optimization using hybrid grey wolf optimization for feature selection, IEEE Access 7 (2019) 39496–39508, http://dx.doi.org/10.1109/ACCESS.2019.2906757.

[58] S.K. Gu, R. Cheng, Y.C. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, Soft Comput. 22 (2018) 811–822, http://dx.doi.org/10.1007/s00500-016-2385-6.

[59] Y. Zhang, D.W. Gong, Y. Hu, W.Q. Zhang, Feature selection algorithm based on bare bones particle swarm optimization, Neurocomputing 148 (2015) 150–157, http://dx.doi.org/10.1016/j.neucom.2012.09.049.