ITC 3/51
Information Technology and Control
Vol. 51 / No. 3 / 2022
pp. 575-591
DOI 10.5755/j01.itc.51.3.31271

A Neighborhood Based Particle Swarm Optimization with Sine Co-sine Mutation Operator for Feature Selection

Received 2022/04/26 | Accepted after revision 2022/08/17

http://dx.doi.org/10.5755/j01.itc.51.3.31271

HOW TO CITE: Qiu, C. (2022). A Neighborhood Based Particle Swarm Optimization with Sine Cosine Mutation Operator for Feature Selection. *Information Technology and Control*, 51(3), 575-591. http://dx.doi.org/10.5755/j01.itc.51.3.31271

A Neighborhood Based Particle Swarm Optimization with Sine Co-sine Mutation Operator for Feature Selection

Chenye Qiu

School of Information Engineering, Huangshan University, No.39 Xihai Road Huangshan China;
phone: +86-0559-2540602; fax: +86-0559-2540602; e-mail: qiuchenye@foxmail.com

Corresponding author: qiuchenye@foxmail.com

Feature selection plays a crucial role in many data mining and machine learning applications. It aims to get rid of those unnecessary features and improve the performance of the classification model. In this paper, a neighborhood based particle swarm optimization with sine cosine mutation operator (NPSOSC) is proposed to select the most informative feature subset. The improvements are included to strengthen its search capacity and avoid local optima stagnation. A distance and fitness based neighborhood search strategy is developed to form stable neighborhood structures for the particles. Each particle adopts superior information from its neighborhoods and the entire swarm can search different regions of the entire search space. The second improvement incorporates a sine cosine mutation operator to enhance the exploration ability and add more randomness into the search process. The improvements will lead to an enhanced balance between exploration and exploitation abilities. To demonstrate the performance of the proposed NPSOSC, seven well-known optimizers are compared with NPSOSC on 16 well-regarded datasets with different difficulty levels. Comparative experiments demonstrate the effectiveness of the proposed NPSOSC in exploring the search space and selecting the most informative features. The statistical test proves the superiority of NPSOSC over other methods is significant.

KEYWORDS: Feature selection, particle swarm optimization, neighborhood search strategy, sine cosine mutation operator.

1. Introduction

With the rapid development of information techniques, huge amounts of data collected in many fields such as social media, industry, and image processing have high dimensions. However, most of the raw datasets contain irrelevant or redundant features which may improve the computational cost and decrease the accuracy of the classification model [4]. Feature selection aims at reducing the dimensionality of the raw datasets while preserving the valuable information as much as possible. Before training the model, feature selection can effectively reduce the training time and enhance the classification performance [36]. Furthermore, it can provide better interpretation of the classification model. Therefore, feature selection is inevitably important in the classification problems with a large number of features.

Feature selection can be viewed as a process of searching for the optimal feature subsets from the raw feature pool. Feature selection methods can be generally divided into two groups: wrappers and filters [18, 19]. A classification model is needed in the wrapper framework to calculate the classification accuracy of a candidate feature subset. On contrary, the filter approach evaluates the importance of the features according to their statistical properties, such as *F*-score criterion, mutual information, and information gain [8, 42]. Generally speaking, wrappers can achieve better classification performance due to the evaluation criterion, but wrappers cost much more computational time than filters since they need to train the classifier repeatedly in the learning process. Filters have the advantage of computational efficiency, but relatively low classification accuracy. Therefore, some hybrid methods are proposed to combine the advantages of the two approaches [30].

Feature selection is a combinatorial *NP*-hard optimization problem due to the large search space. For a dataset with m features, the number of possible feature subsets is 2^m. Furthermore, the interactions among the features should not be neglected. An irrelevant feature might be very informative with the existence of another feature. Also, a useful feature may become redundant when some other features are included. The search strategy in feature selection can be categorized to three groups: exhaustive search, heuristic search, and meta-heuristic methods [26]. In most of the cases, exhaustive search method is impractical because of the large search space. Heuristic strategy is much more computation efficient but it cannot explore the entire feature space effectively.

More recently, nature inspired meta-heuristic algorithms have been applied to feature selection problem, including genetic algorithm (GA) [35], particle swarm optimization (PSO) [38], forest optimization algorithm [40], grey wolf optimizer [23], and sine cosine algorithm [24]. These meta-heuristic algorithms show powerful global search ability and computational efficiency in high-dimensional optimization problems. Among various meta-heuristic algorithms, PSO is known for its fast convergence speed and ease of implementation and it has gained much attention in solving various real-world optimization tasks [15, 29]. In PSO, each particle represents a candidate solution to the optimization problem and a set of particles traverse the entire search space to locate the global optimal position. Each particle modifies its search trajectory according to its own flying experience and the best experience of the population. PSO has numerous advantages in various aspects, such as fewer control parameters, computational efficiency, and fast convergence speed.

Although PSO based feature selection methods have shown promising results [27], there are still several shortcomings with the algorithm. In PSO, the global best (*gbest*) plays an important role in guiding the entire swarm, but this may be an obstacle in the datasets with multiple local optima. The *gbest* learning mechanism has a negative effect on keeping the population diversified. Moreover, if the current *gbest* is located near a local optimal solution, other particles move towards the *gbest* and it is difficult for them to jump out of local optima. Besides, PSO facilitates fast convergence but its exploration ability is relatively weaker than other meta-heuristics. These shortcomings would limit the performance of PSO in high-dimensional datasets [27]. Therefore, further research is still needed to develop more promising PSO based feature selection models. Motivation by these reasons, this paper proposes a novel neighborhood based PSO with sine cosine mutation operator (NPSOSC) to select the most informative feature subsets. A distance and fitness based neighborhood search strategy is used to build stable and appropriate neighborhoods for the particles and each particle learns superior information from its neighboring particles instead of the *gbest*. The neighborhood search strategy can explore more candidate feature subsets in the entire search space and find the best feature subset in high-dimensional feature space. Moreover, a sine cosine mutation operator is introduced to PSO to strengthen its global search ability and enrich its search behavior. The main contributions of this paper include:

1. A distance and fitness based neighbor-hood search strategy is proposed to substitute the *gbest* learning mechanism in PSO. Each particle adopts valuable information from its neighbor-hoods to update its position. The novel search strategy can enhance the search capacity of PSO in high-dimensional objective space and keep the population diversified during the search process.
2. A sine cosine mutation operator is incorporated to PSO to strengthen its global exploration capacity and improve the performance of the algorithm in high-dimensional datasets. Furthermore, it can lead to a better balance between the exploration and exploitation abilities.
3. The proposed NPSOSC is applied to solve feature selection problem. Computational experiments and comparisons on 16 well-regarded datasets demonstrate that the proposed NPSOSC can select the most valuable feature subsets and outperform some other meta-heuristic based feature selection models.

The rest of the paper is structured as follows. Section 2 provides a brief review of the classical PSO and reviews the related researches. In Section 3, a neighborhood PSO with sine cosine mutation operator for feature selection is presented. Section 4 presents the experimental results and discussions. The conclusions and future work are outlined in Section 5.

2. Background and Related Works

2.1. Particle Swarm Optimization

Particle swarm optimization is an optimization algorithm based on swarm intelligence, which is motivated by swarm behavior such as bird flocking and fish schooling [15]. Compared with other meta-heuristic methods, PSO has fewer adjustable parameters and shows fast convergence speed. Therefore, it has been extensively researched and has been successfully applied to a wide range of real world optimization problems.

PSO works by maintaining a swarm of particles and each particle represents a candidate solution of the optimization problem. Particles fly in a multi-dimensional space to search for the optimal position. At the beginning of the algorithm, all the particles are randomly initialized to spread the particles uniformly in the feasible search space. A fitness function is used to evaluate the quality of the particles. The best position that a partcile has located is its own personal best (*pbest*) and the optimal position explored by the entire population is called the *gbest*. In the search process, each particle updates its velocity and position with the guidance of the *gbest* and its own *pbest*.

Let $X_i = (x_1, x_2, \cdots, x_D)$ denote the position of the *i*th particle in the swarm. D is the dimension of the search space. Its current velocity is $V_i = (v_1, v_2, \cdots, v_D)$. In the canonical PSO algo-rithm, the velocities and positions of particles are updated by the following equations:

$$\begin{aligned} V_i^{t+1} &= w \times V_i^t + c_1 \times r_1 \times (pbest_i - X_i^t) \\ &\quad c_2 \times r_2 \times (gbest - X_i^t) \end{aligned} \tag{1}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}, \tag{2}$$

where V_i^t and X_i^t denote the *i*th particle's velocity and position in iteraion t, respective- ly; $pbest_i$ is the *i*th particle's *pbest*; w is the inertia weight which is very crucial in PSO. A larger inertia weight facilitates global exploration while a smaller one is beneficial for local exploitation. The most popular strategy is to use a linear decresing inertia weight. c_1 and c_2 are the cognitive and social weights, respectively; r_1 and r_2 are uniformly randomized between [0,1].

2.2. Binary PSO for Feature Selection

Feature selection model contains an evaluator and a search method. When PSO is used as the search method, some operator needs to be modified. PSO is originally developed to search in continuous objective space. Therefore, the positions and velocities of the particles are described by real numbers, but feature selection is a discrete optimization problem.

Many researchers employed the Binary PSO (BPSO) [16] to solve feature selection problem. When using BPSO for feature selection, the position of particle is restricted to 1 or 0. 1 means the corresponding feature is chosen. 0 denotes the feature is excluded. The velocity means the probability of the feature being chosen. Initially, the position and velocity of each particle are generated as follows:

$$x_{id}^{t} = \begin{cases} 1, \text{ if } rand() > 0.5 \\ 0, otherwise \end{cases} \tag{3}$$

$$v_{id}^{t} = -v_{\max} + 2 \times rand() \times v_{\max}, \tag{4}$$

where *rand*() is a random number uniformly distributed in [0,1]. v_{max} is pre-defined parameter which is used to limit the maximum speed of each particle. It is an important variable in BPSO. Each particle updates its velocity according to Equation (1). The position updating mechanism is expressed as follows:

$$x_{id}^{t} = \begin{cases} 1, \text{ if } S(v_{id}^{t}) > rand() \\ 0, otherwise \end{cases}, \tag{5}$$

where *S*() is a sigmoid limiting transformation function and it is defined as:

$$S(V_{id}^{t}) = \frac{1}{1+\exp(-V_{id}^{t})}. \tag{6}$$

BPSO maintains the main mechanism of PSO and it has been widely used in various discrete optimization problems. Except the BPSO, the canonical PSO can also be applied to feature selection with only one small variation. In this method, each particle updates its velocity and position vectors the same as the continuous PSO, using Equations (1)-(2). Before calculating the fitness value, the position vector of each particle is decoded to translate real numbers into a binary string. The position of a particle is restricted between [0,1]. If the value is larger than a predefined threshold (e.g. 0.5), it means the corresponding feature is chosen. Previous researches have proved that this method can achieve similar or even better performance than BPSO in feature selection [18].

2.3. Related Works

Wrapper approaches aim at reducing the dimensionality of the datasets which is vital in many practical applications. The wrapper approach employs a classification model to assess the quality of the feature subset. It is argued to have better classification performance than the filter approach, but it costs much computational time and may over-fit the training dataset. Various meta-heuristic based wrapper models have been developed.

Raymer et al. [35] utilized GA to select optimal feature subset and compared it with two traditional feature selection methods. Maleki et al. [22] employed GA to select useful feature subsets in the lung cancer prognosis problem. GWO has been used as a wrapper feature selection in [9]. In [1], the performance of GWO in feature selection was strengthened by introducing a two-phase mutation operator. Authors in [2] proposed a chaos-based salp swarm algorithms (SSA) to select optimal feature subset. Mafarja and Mirjalili [21] developed a feature selection model base on the whale optimization algorithm (WOA). Authors in [43] proposed a gaussian mutational chaotic fruit fly optimization algorithm (MCFOA) to select optimal feature subset. Self-adaptive cohort intelligence (SACI) was used to generate optimal feature subsets in [3]. Binary variants of grasshopper optimization algorithm (GOA) were developed to solve feature selection task in [20]. Arora et al. [5] applied butterfly optimization algorithm (BOA) to handle feature selection problem for classification purpose. Zorarpac et al. [44] applied the hybridization of artificial bee colony (ABC) and differential evolution (DE) for feature selection problem. This method consists of a new binary neighborhood search mechanism, a novel modified onlooker bee process, and a differential mutation operator and it outperforms ABC and DE. Hu et al. [13] proposed an enhanced version of black widow optimization algorithm (BWO) for the feature selection problem.

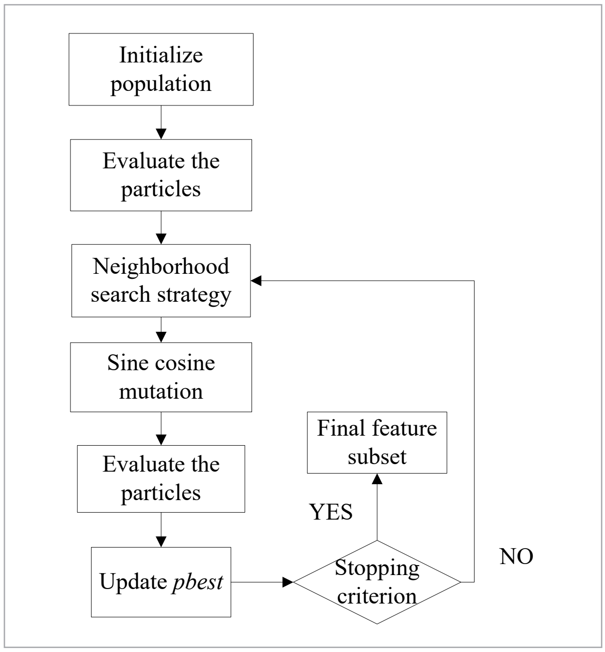
PSO is a powerful stochastic search algorithm with strong global search ability and it has gained much attention in feature selection problem. Xue et al. [41] developed a PSO based feature selection model with novel initialization and updating mechanisms. Moradi et al. [25] combined PSO with a local search operator to generate smaller feature subsets. The local search operator can help PSO select distinct features by considering the correlation information. Tran et al. [39] proposed a PSO approach with a new particle representation scheme to deal with the feature selection problem. Barebones PSO (BBPSO) was used to select optimal feature subset in [32] and a novel chaotic jump operator was developed to enrich the search behavior and add more randomness into the search process. Gu et al. [11] introduced a new variant of PSO, called competitive swarm optimization (CSO), to solve high-dimensional feature selection problem. Engelbrecht et al. [10] employed the set based PSO for feature selection in which a set based particle representation was used and the particle updating mechanisms were redefined. Tran et al. [38] introduced a variable-length PSO to solve feature selection problem which can enhance the search ability and generate smaller feature subsets. In [31], PSO with multi-swarm topology was developed for obtaining optimal feature subset in which several sub-swarms evolved simultaneously. Chen et al. [8] hybridized PSO with a spiral-shaped method for feature selection problem. Thaher et al. [37] proposed a binary PSO boosted with evolutionary population dynamics for feature selection in which the worst particles were replaced by the high quality solutions around them. Qiu et al. [33] proposed a PSO based feature selection model using a novel three layer structure which can improve the population diversity. Hu et al. [14] proposed a multi-surrogate assisted binary particle swarm optimization for feature selection which was able to reduce running time and prediction error. Asif et al. [6] developed a feature selection model based on a self-inertia weight adaptive PSO and the feature selection model was used for text classification.

From the literature survey, it can be seen that the performance of PSO in feature selection is still far from perfect. PSO still suffers from local optima stagnation and premature convergence in high-dimensional datasets. The performance of PSO in large scale datasets is not stable since its performance depends on the particular datasets and the search strategy. To address the existing shortcomings, a novel neighborhood based PSO with sine cosine mutation operator is proposed to solve feature selection problem in this study.

3. The Proposed Approach

This section describes the NPSOSC for feature selection. In the canonical PSO, each particle updates its position by learning from its own *pbest* and the *gbest* found by the whole swarm. This strategy may lead to

Figure 1
Flowchart of the proposed NPSOSC



unsatisfactory results in high-dimensional datasets. A distance and fitness based neighborhood search strategy is proposed in this paper, in which each particle utilizes the superior local information to guide its search process instead of the *gbest*. Furthermore, a sine cosine mutation operator is employed to improve the global search ability. Figure 1 shows the flowchart of the proposed NPSOSC. The details of the key components will be introduced later in this section.

3.1. Particle Representation

PSO is predominately used for dealing with continuous optimization problems. Since feature selection is a combinatorial optimization problem, the canonical PSO should be adapted to address large-scale feature selection problems. In this study, the continuous PSO is employed since previous research demonstrates this method can work effectively in discrete search space with some pre-determined decoding scheme [41, 31].

For particle i, its position vector is denoted as $X_i = (x_1, x_2, \cdots, x_D)$. D is the dimension of the search space, i.e., the raw feature number of the dataset. Initially, the position vector is coded in real number in the range of [0,1]. Next, a threshold is used to transfer the original vector into a binary string. The conversion from real numbers to a binary string is performed as follows:

$$A_{id} = \begin{cases} 1, \ if \ x_{id} > \theta \\ 0, \ otherwise \end{cases}, \quad (7)$$

where A_{id} is the obtained candidate feature subset. θ is the threshold which is set to 0.5 in this work. A_{id}=1 means the dth feature is selected. Otherwise, this feature is not included in this feature subset. By using this particle representation and decoding scheme, the canonical PSO can work effectively in feature selection problem and keep the main structure and advantages of PSO.

3.2. Fitness Function

This study aims at building a PSO based wrapper model and a classification algorithm is needed to calculate the classification accuracies of the feature subsets. Compared with other new and powerful learning algorithms [17], the K nearest neighbor algorithm (KNN) is employed due to its simplicity and robustness. KNN has been widely used in many practical machine learning and data mining tasks. Users only need to set the value of K in this algorithm. The fitness function is expressed as follows:

$$\min \ f(x_i) = error + \lambda \times \frac{num_s}{D}, \quad (8)$$

where *error* denotes the classification error rate obtained by KNN. num_s is the feature subset size. λ is a parameter to control the importance of classification performance and feature reduction. λ is set as 0.1 which is a suitable tradeoff between error rate and selected feature number. According to Equation (8), this fitness function can lead the algorithm to search for accurate feature subsets with fewer features.

3.3. Neighborhood Search Strategy Based on Distance and Fitness

As shown in Equation (1), *gbest* plays a vital role in the process of searching for better solutions. Under the guidance of the *gbest*, the entire swarm would converge to the vicinity area of the *gbest*. Therefore, the algorithm suffers from the quick loss of population diversity and premature convergence, especially in high-dimensional feature selection problems. To avoid the defect, a neighborhood based PSO is employed for feature selection. The velocity of particle i is updated as follows:

$$V_i^{t+1} = w \times V_i^t + c_1 \times r_1 \times (pbest_i - X_i^t) + c_2 \times r_2 \times (lbest_i^t - X_i^t), \quad (9)$$

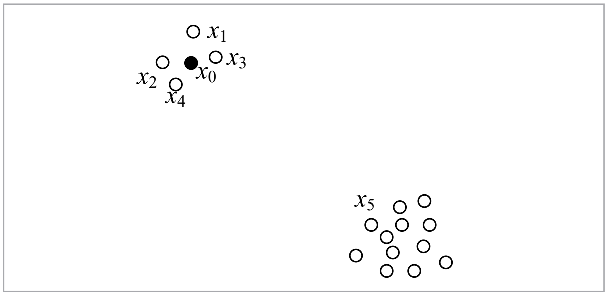
where $lbest_i^t$ denotes the local leader for particle i in the tth iteration. In the *lbest* PSO, the population is divided into many sub-populations and each particle adopts valuable information from its neighboring particles instead of learning from the unique *gbest*. Hence, the *gbest* would not dominate the search process of the algorithm. The particles can explore more regions with the guidance of its own *lbest* which is beneficial for maintaining population diversity and preventing the algorithm from falling into local optima.

However, it is difficult to form stable and appropriate neighborhoods for particles [12]. Traditional methods usually adopt a fixed sized neighborhood and the optimal particle in the neighborhood set is selected to be the local leader [34]. If the number of neighbors is not properly set for the swarm distribution, some unsuitable particles might be chosen as the local leader and this will have a negative effect on other particles [28]. Figure 2 shows a simple 2-dimensional problem. Assume the neighborhood size is 5. In Figure 2, most of the particles locate at the southeast corner of the search space and the current particle x_0 lies in the northwest corner. According to the Euclidean distance, the neighborhood set of x_0 contains x_1, x_2, x_3, x_4, and x_5. If x_5 owns better fitness value than other particles, x_0 will leave its current vicinity area by the guidance of x_5. In this situation, particle i cannot search its vicinity area efficiently and the neighborhood structure is broke. This is not helpful in maintaining population diversity and searching its neighboring area sufficiently. Moreover, if x_5 corresponds to a local optimal solution, it will guide other particles move towards the local optimum. Therefore, how to form a stable neighborhood structure is crucial in the *lbest* PSO.

In order to deal with the problem, this work proposes a novel distance and fitness based neighborhood strategy to choose appropriate learning exemplar for each particle. For particle i, find its m nearest par-

Figure 2

Illustration of a simple 2-dimension space



ticles according to the Euclidean distance and its neighborhood set is shown as:

$$NBS_i = \{x_{i1}, x_{i2}, \cdots, x_{im}\}. \tag{10}$$

For the m particles, they are sorted in ascending order according to the distance. Afterwards, the neighborhood set is sorted according to an increasing order of the particles' fitness values. Each particle j in the neighborhood set has a distance rank DR_j and a fitness rank FR_j. Then, the two ranks are integrated into the final rank according to:

$$R_j = \beta \times DR_j + (1-\beta) \times FR_j, \tag{11}$$

where β is the weighting factor. By using Equation (11), the final rank can achieve a balance between distance and fitness value. In this study, the weighting factor is set to 0.6.

Take Figure 2 for example, the distance rank for the 5 particles in the neighborhood set is 4, 3, 2, 1, and 5, respectively. x_5 owns the best fitness value and the fitness rank of other particles are randomly set, so we have the fitness rank 3, 2, 5, 4, and 1, respectively. According to the traditional method, x_5 will be chosen as the local leader since it owns the best fitness value. However, in the proposed method, the final rank is obtained by considering both fitness value and distance, so the final rank is 3.6, 2.6, 3.2, 2.2, and 3.4, respectively. Hence, x_4 is selected as the *lbest*. By using the novel neighborhood strategy, it can ensure that inappropriate particles will not be chosen as local leaders.

Compared with the traditional neighborhood search method, the proposed approach considers both fitness value and distance in selecting the *lbest*. By using the weighting method, those relatively remote particles with good fitness values would not be chosen as the *lbest*. Therefore, it is likely to retain a more stable neighborhood structure and the particles can search in its neighboring areas more sufficiently. The neighborhood size is set to 5 in this study. The pseudocode of the neighborhood search strategy is shown in Algorithm 1.

Algorithm 1. Distance and fitness based Neighbor-hood Search Strategy

1: **For** each particle i **do**

2: Calculate Euclidean distance between x_i and other particles in the swarm;

3: Sort the distance in ascending order;

4: Choose the top m particles to build the neighborhood set;

5: Store the particles in the neighborhood set according to fitness value in ascending order;

6: Obtain the final rank of the neighboring particles with Equation (11);

7: Choose the highest rank particle in the neighborhood set as the *lbest*;

8: Update the velocity with Equation (9);

9: Update the position with Equation (2);

10: **End**

3.4. Sine Cosine Mutation Operator

The convergence speed of the original *lbest* PSO is relatively poor, especially in high-dimensional optimization problems. In this work, a sine cosine mutation operator is developed to improve the exploration ability of the algorithm and speed up its convergence. SCA is a relatively new population based intelligent algorithm developed by Mirjalili [24]. SCA uses the characteristics of sine and cosine trigonometric functions to realize exploration and exploitation and it has strong global search ability. Therefore, a sine cosine mutation operator is introduced to the neighborhood based PSO. As is shown in Figure 1, the sine cosine mutation operator is conducted on the whole swarm after the position updating process. For each particle i, a mutation rate is used to decide whether to perform the mutation operator. For each dimension of the particle i, the new position is generated by:

$$x_{new}^{t+1} = \begin{cases} x_i^t + r_1 \sin(2\pi r_2) \left| 2r_3 x_{gbest} - x_i^t \right|, & if\ r_4 < 0.5 \\ x_i^t + r_1 \cos(2\pi r_2) \left| 2r_3 x_{gbest} - x_i^t \right|, & otherwise \end{cases}, \tag{12}$$

where r_2, r_3, and r_4 are random numbers in the range (0,1). r_4 controls the switch between the sine and cosine functions which is beneficial for improving randomness between exploration and exploitation. r_1 is the transition parameter which decides the search space around the current solution. The parameter r_1 provides a smooth transition from the phase of exploration to exploitation. In SCA, r_1 linearly decreases from 2 to 0. In the first half of the search process, large values of r_1 contribute in the exploration of the search space, after which smaller values of r_1 facilitate exploitation.

In this paper, the sine cosine operator is used as a mutation operator in the neighborhood based PSO. The mutation rate is set to 0.2 in this work. This operator can bring more randomness into the search behavior and improve the possibility of escaping from local optimal solutions. This sine cosine mutation operator can explore more regions in the search space and it would not bring additional fitness evaluations. Hence, the proposed NPSOSC is expected to achieve a comparatively better balance between population diversity and convergence speed.

3.5. The Framework of NPSOSC

The framework of NPSOSC is shown in Algorithm 2.

Algorithm 2. The pseudocode of NPSOSC

1: Initialize the population randomly;

2: Evaluate the particles with Equation (11);

3: Initialize the *pbest* of each particle;

4: **For** each iteration **do**

5: **For** each particle *i* **do**

6: Update its velocity and position according to Algorithm 1;

7: Perform the sine cosine mutation operator with a certain probability;

8: Evaluate its fitness value with Equation (11) and update its *pbest*;

9: **End**

10: **End**

11: Calculate the classification accuracy of the optimal dataset in the test dataset;

12: Return the optimal feature subset and its classification accuracy in the test set.

4. Experiment

4.1. Datasets

To testify the effectiveness of the proposed NPSOSC, 16 datasets are considered for simulation experiments. The first 14 of them are taken from the UCI (University of California, Irvine) machine learning repository and the remaining two are gene expression datasets with a large number of features. These datasets show considerable diversity over dimensionality, size and number of classes. These datasets have been widely used in many feature selection studies [5, 9, 41]. The details of these datasets are shown in Table 1.

Each dataset is randomly divided into two parts. 70% of the instances are used as the training set and the remaining 30% instances form the test set. In the training process, KNN is employed to compute the classification performance of the feature subsets. For each dataset, the classification accuracy of KNN is calculated through a 10-fold cross validation. The detailed process of the 10-fold cross validation can be found in [25, 31]. Because Colon and Leukemia data-

Table 1
Datasets

Dataset	Features	Instances	Classes
Glass	9	214	6
Heart	13	270	2
Wine	13	178	3
Australia	16	690	2
Zoo	16	101	6
Lymphography	18	248	3
Spect	22	267	2
Parkinson	22	195	2
WBCD	30	569	2
Ionosphere	34	351	2
Sonar	60	208	2
Hillvalley	100	606	2
Musk1	166	476	2
LSVT	309	126	2
Colon	2000	62	2
Leukemia	7129	72	2

sets contain relatively small number of instances, 5-fold cross-validation is applied to the two sets. After the training process, the obtained feature subsets are evaluated in the test set using KNN. The number of neighbors in KNN is 5 in this study.

4.2. Comparative Algorithms and Parameter Settings

To validate the performance of the proposed algorithm, it is compared with seven meta-heuristics based wrapper approaches, including: PSO [15], SCA [24], GOA [26], GWO [9], CSO [11], BOA [5], and WOA [21]. For all the involved approaches, the number of search agents and iterations are set to 20 and 50, respectively. Other specific parameter settings for each optimizer are outlined in Table 2. To remove the influence of random factors, all feature selection methods are run 30 independent times on all the datasets. The experiments are all performed on a machine with Intel(R) Core(TM) i5-6500 at 3.2 GHz and 8.00 GB of RAM using MATLAB. The operating system is MS Windows 10.

Table 2

Parameter settings for each optimizer

Algorithm	Parameter	Value
PSO	c_1	2
	c_2	2
	w	[0.9 0.4]
SCA	r_1	[2,0]
GOA	f	0.5
	l	1.5
GWO	a	[2 0]
CSO	Social factor	0.1
BOA	a	0.1
	c	[0.01 0.25]
WOA	a	[2 0]

To demonstrate the effectiveness of NPSOSC, the following sets of experiments are conducted:

1. NPSOSC is compared with seven meta-heuristic methods is terms of classification accuracy, feature subset size, and computational time.
2. The non-parametric Wilcoxon Rank Sum is used to verify the significance level of two feature selection methods.
3. Convergence curves are drawn to show the convergence speed and precision of NPSOSC.
4. NPSOSC is compared with the traditional neighborhood search strategy to show its efficacy in feature selection problem.

4.3. Results of Comparative Experiments

NPSOSC is compared with other meta-heuristic based wrapper approaches on the 16 datasets. The efficacy of these feature selection methods are evaluated using the classification performance measure, feature subset size, statistical metric, and running time.

Table 3 reports the mean classification accuracies and standard deviations of the eight algorithms in 30 independent runs. For each dataset, the highest mean classification accuracy is shown in **boldface**. According to Table 3, it can be revealed that the proposed NPSOSC achieves the highest mean classification accuracy in 9 out of 16 datasets. For example for the Ionosphere dataset, the mean accuracy NPSOSC is reported 0.8732 while CSO places 2nd with 0.8572 and GWO ranks 3rd with 0.8569. PSO-NSSC places 2nd in four datasets and 3rd in three datasets, but the performance of NPSOSC is very close to the leaders in these seven sets. Take the Spect dataset for example, the accuracy of NPSOSC is 0.7903 while the value of the leader is 0.7926. The gap between them is relatively small. In the five high-dimensional (>100 features) datasets, NPSOSC achieves the best results in three datasets and places 2nd in the other two datasets. Moreover, NPSOSC obtains lower standard deviations in the Musk1 and Colon datasets which indicate the performance of NPSOSC is more stable. The last two rows of Table 3 show the mean and final rank of the eight algorithms in all the 16 datasets. The mean rank of a specific algorithm is calculated by averaging its ranks in all the 16 datasets. In terms of the final rank, NPSOSC places 1st with the value of 1.63, while GOA (4.38) and SCA (4.56) place 2nd and 3rd, respectively. The results of Table 3 indicate that NPSOSC achieves better accuracy than other methods. The two improves of NPSOSC play a vital role in improving the search capacity of the algorithm.

Table 4 outlines the average feature subset size ob-

Table 3

Classification accuracies and standard deviations obtained by different algorithms

	PSO	SCA	GOA	GWO	CSO	BOA	WOA	NPSOSC
Glass	0.6636	**0.6838**	0.6405	0.6338	0.6354	0.6354	0.6538	0.6815
	0.0707	0.0429	0.0143	0.0107	0.0299	0.0146	0.0477	0.0492
Heart	0.8296	0.8444	0.8232	0.8339	0.8247	0.8358	0.8173	**0.8523**
	0.0152	0.0268	0.0288	0.0319	0.0312	0.0377	0.0278	0.0213
Wine	0.9704	0.9667	0.963	0.963	0.9704	0.9667	0.9463	**0.9827**
	0.0166	0.0078	0.0175	0.0147	0.0156	0.017	0.0238	0.0144
Australia	0.8423	0.8334	0.8378	0.826	0.8385	0.8351	**0.8451**	0.8446
	0.017	0.0365	0.0174	0.0232	0.0189	0.0219	0.0145	0.0095
Zoo	0.9204	0.9226	0.9339	**0.9516**	0.9355	0.9452	0.8903	0.9403
	0.0269	0.0212	0.0214	0.0222	0.0263	0.0306	0.0531	0.027
Lymphography	0.7378	0.7382	**0.7767**	0.7689	0.7422	0.7522	0.7644	0.7762
	0.0266	0.0342	0.0385	0.0335	0.0459	0.0403	0.0447	0.0416
Spect	0.7877	0.7879	0.777	**0.7926**	0.7802	0.784	0.791	0.7903
	0.0152	0.0188	0.0268	0.0113	0.0278	0.0145	0.014	0.0134
Parkinson	0.8881	0.8878	0.8878	0.8898	0.9017	0.8983	0.8949	**0.9089**
	0.0164	0.0184	0.022	0.0268	0.0107	0.0309	0.0274	0.0132
WDBC	0.9587	0.9579	0.9612	0.955	0.9564	0.9602	0.9591	**0.9628**
	0.0126	0.0096	0.0132	0.0163	0.0096	0.0106	0.0143	0.0109
Ionosphere	0.8491	0.8368	0.8569	0.8316	0.8572	0.8264	0.8453	**0.8732**
	0.0204	0.0209	0.0263	0.033	0.0284	0.0205	0.0173	0.0191
Sonar	0.7926	0.771	0.8012	0.7992	0.7746	0.7905	0.7921	**0.8177**
	0.0326	0.0419	0.0348	0.0283	0.0288	0.0316	0.0264	0.0248
Musk1	0.8392	0.8492	**0.8512**	0.8287	0.8352	0.8182	0.8126	0.8485
	0.0208	0.0161	0.0234	0.0219	0.0278	0.0255	0.0221	0.0147
Arrhythmia	0.6653	0.6672	0.6658	0.6507	0.6763	0.6596	0.6551	**0.7022**
	0.0187	0.0144	0.012	0.0241	0.0156	0.018	0.0231	0.0117
LSVT	0.7421	0.7474	0.7579	0.7066	0.7368	0.7579	0.7553	**0.7893**
	0.0461	0.0544	0.0643	0.0477	0.0411	0.0688	0.0448	0.0393
Colon	0.7579	**0.7947**	0.7543	0.7684	0.7579	0.7632	0.7842	0.7918
	0.0368	0.0461	0.0467	0.0272	0.0248	0.0372	0.0461	0.0223
Leukemia	0.6823	0.6836	0.6853	0.6381	0.6727	0.7109	0.6682	**0.7122**
	0.0246	0.0263	0.0225	0.0161	0.0162	0.0213	0.0231	0.0164
Mean rank	4.88	4.56	4.38	5.56	5.06	4.81	5.13	**1.63**
Final rank	5	3	2	8	6	4	7	**1**

Table 4

Average number of selected features obtained by different algorithms

	PSO	SCA	GOA	GWO	CSO	BOA	WOA	NPSOSC
Glass	2.87	2.9	2.75	2.95	3	3	**2.7**	3.3
Heart	4	4.5	4.32	5.55	4.7	4.5	4.3	**3.9**
Wine	5.1	5.1	4.9	5.5	**4.4**	5.4	4.6	4.65
Australia	5.27	4	5.1	4.7	3.8	3.5	**3**	3.7
Zoo	7.73	7.4	7.1	8.95	8.2	7.7	8.3	**7.2**
Lymphography	**6.3**	6.8	8.2	8.8	7.3	7.1	9.4	6.75
Spect	9.9	8.7	8.2	11.5	**6.6**	10.1	11.1	8.15
Parkinson	7.9	6.9	6.9	10.3	6.8	9.1	9.3	**6.7**
WDBC	12.2	9.8	9.75	11	**6.6**	10	8.9	8.6
Ionosphere	7.9	**7.2**	8.4	13.25	7	11.1	9.7	7.4
Sonar	23.2	24.45	24.1	33.15	22.2	26.8	25.3	**22.05**
Musk1	78.47	74.1	72.8	94.7	**68.8**	75.3	71.1	71.8
Arrhythmia	124.4	127.8	126.5	165.5	119.4	129.8	120.3	**117.2**
LSVT	143.7	149.9	147.2	195.65	**135.9**	143.3	153.9	136.35
Colon	963.7	969.6	950.8	1244.4	886.9	939.5	936.7	**819.05**
Leukemia	3485.4	3432.8	3332.8	3660.8	3186.6	3416.9	2550.9	**2120.7**
Mean rank	4.75	4.69	4.19	7.56	2.81	5.44	4.25	**2.31**
Final rank	6	5	3	8	2	7	4	**1**

tained by the eight algorithms. For each dataset, the smallest average feature subset size is shown in **boldface**. Table 4 shows that all the feature selection algorithms can effectively reduce the dimensionality of the raw datasets. Among all the methods, NPSOSC produces the smallest feature subsets in seven datasets. It is worth mentioning that NPSOSC selects much fewer features in the two datasets with thousands of features, i.e, Colon and Leukemia. According to the final rank, the best algorithm is NPSOSC (2.31) while CSO and GOA place 2nd (2.81) and 3rd (4.19), respectively.

It can be concluded from Tables 3-4 that the proposed PSO-NSSC is superior to other wrapper based feature selection methods. NPSOSC can effectively improve the classification accuracy and reduce the number of features. Take the Colon dataset for example, the accuracy of SCA is a bit better than NPSOSC, but NPSOSC has a substantial superiority over SCA in feature reduction in this dataset. On a whole, the experimental results suggest that NPSOSC is able to generate high quality feature subsets.

To verify whether there is statistical difference between the classification accuracies of NPSOSC and those comparative approaches, the non-parametric Wilcoxon Rank Sum test is performed at 5% significance level. If the p-value is less than 0.05, there is a significant difference. The results of the statistical test are outlined in Table 5. Here, '+/=/-' denote the performance of NPSOSC is significantly better than, almost consistent to, and significantly worse than its comparative approach respectively. In all the 16 datasets, NPSOSC obtains significantly better or consistent results compared with other feature selection approaches. The Wilcoxon test implies that the proposed NPSOSC shows superior performance over other seven methods in feature selection problems.

Table 6 outlines the mean computational time (in seconds) in 30 independent runs of the eight algorithms in the 16 datasets. The shortest computational time in

Table 5
Results of the Wilcox test of NPSOSC vs other algorithms

	PSO	SCA	GOA	GWO	CSO	BOA	WOA
Glass	=	=	+	+	+	+	+
Heart	+	=	+	=	+	=	+
Wine	+	+	+	+	=	+	+
Australia	=	+	=	+	=	+	=
Zoo	+	+	=	=	=	=	+
Lymphography	+	+	=	=	+	+	=
Parkinson	+	=	+	+	=	=	=
Spect	=	=	=	=	=	=	=
WDBC	+	+	=	+	+	=	+
Ionosphere	=	=	=	=	=	=	=
Sonar	=	=	=	=	+	+	+
Musk1	=	=	=	+	=	+	+
Arrhythmia	+	+	+	+	+	+	+
LSVT	+	+	=	+	+	=	=
Colon	+	=	+	=	+	=	=
Leukemia	+	+	+	+	+	=	+
+/=/-	10/6/0	8/8/0	7/9/0	9/7/0	9/7/0	7/9/0	9/7/0

Table 6
Running time of different algorithms (in seconds)

	PSO	SCA	GOA	GWO	CSO	BOA	WOA	NPSOSC
Glass	4.63	4.68	4.96	7.82	4.32	4.39	**4.08**	4.49
Heart	14.3	14.94	10.66	13.59	10.42	13.49	**8.31**	9.98
Wine	7.56	7.93	5.65	7.01	**4.5**	6.31	4.95	7.95
Australia	31.88	35.74	37.18	**31.42**	47.38	34.69	34.63	35.55
Zoo	2.03	2.09	2.05	3.05	2.06	2.15	2.55	**1.86**
Lymphography	4.22	4.27	5.25	5.88	5.21	5.76	5.81	**3.88**
Parkinson	9.05	9.89	10.44	11.98	**8.97**	9.78	11.44	9.41
Spect	12.17	14.16	14.49	13.3	**11.33**	16.52	13.04	12.59
WDBC	28.58	28.84	26.84	28.59	28.72	**25.22**	26.25	29.24
Ionosphere	12.83	12.29	13.32	12.77	19.13	**11.47**	11.8	14.41
Sonar	6.56	6.98	6.39	6.94	6.32	**6.12**	6.43	6.54
Musk1	46.1	47.42	**45.7**	51.54	46.2	43.11	48.93	48.98
Arrhythmia	**57.76**	59.83	59.83	72.25	62.74	60.23	85.88	62.64
LSVT	6.5	6.36	**6.4**	7.77	6.57	6.63	7.73	6.6
Colon	12.12	12.71	11.24	14.66	**10.02**	10.19	13.94	12.7
Leukemia	88.23	90.68	91.68	110.21	**81.2**	84.81	113.43	90.57
Mean rank	3.56	5.19	4.38	6.44	3.5	3.63	4.75	4.56
Final rank	3	6	5	7	1	2	5	4

Table 7
Comparison of different neighborhood strategies

	NPSOSC1		NPSOSC2		NPSOSC1	NPSOSC2
	accuracy	std.	accuracy	std.	# of features	
Glass	**0.6815**	0.0492	0.6617	0.0288	3.3	**3.13**
Heart	**0.8523**	0.0213	0.8507	0.021	**3.9**	4.44
Wine	**0.9827**	0.0144	0.9803	0.0056	**4.65**	5.26
Australia	0.8446	0.0095	**0.8527**	0.0058	**3.7**	4.67
Zoo	0.9403	0.027	**0.9429**	0.0213	7.2	**7.1**
Lymphography	**0.7762**	0.0416	0.7658	0.0316	**6.75**	6.8
Spect	**0.7903**	0.0134	0.7768	0.0156	8.15	**7.9**
Parkinson	**0.9089**	0.0132	0.8876	0.0126	**6.7**	6.9
WDBC	0.9628	0.0109	**0.9662**	0.0068	**8.6**	9.2
Ionosphere	**0.8732**	0.0191	0.8724	0.0229	**7.4**	7.75
Sonar	**0.8177**	0.0248	0.7981	0.0318	**22.05**	27.64
Musk1	0.8485	0.0147	**0.8515**	0.0104	**71.8**	78.82
Arrhythmia	**0.7022**	0.0117	0.6879	0.0123	**117.2**	120.8
LSVT	**0.7893**	0.0393	0.7768	0.0522	**136.35**	146.21
Colon	**0.7918**	0.0223	0.7807	0.0427	**919.05**	924.79
Leukemia	**0.7122**	0.0164	0.7047	0.0278	**3120.7**	3190.1

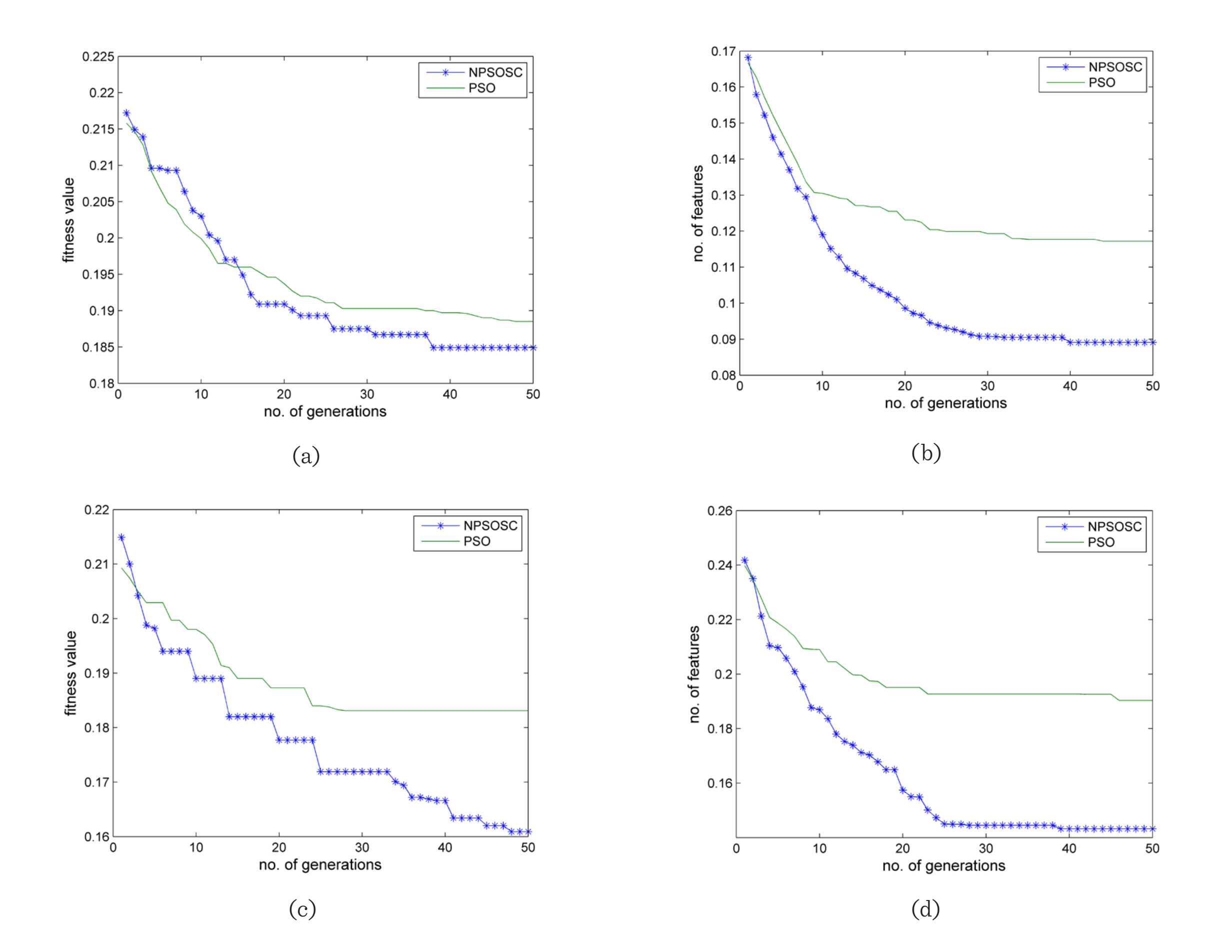
each dataset is shown in **boldface**. In all the involved algorithms, NPSOSC ranks 4th considering the computational time. Compared with the canonical PSO, NPSOSC needs longer time in most of the datasets due to the incorporation of two new operators. Selecting the *lbest* based on fitness value and Euclidean distance is computationally expensive. However, NPSOSC achieves noticeable advantages over PSO in improving classification performance and reducing feature numbers. Hence, there is a trade-off between the quality of obtained feature subsets and the running time. The increased computational cost is acceptable.

4.4. Analysis on the Convergence Curves

Four datasets (Heart, Ionosphere, Musk1, and Colon) are selected as representatives to analyze the convergence process of the proposed NPSOSC and the classical PSO. These datasets cover small, median, and large datasets. The convergence curves of the two methods are plotted in Figure 3. In these figures, x-axes are the iteration numbers and y-axes show the average fitness values in 30 independent runs. The convergence cure can display convergence speed and convergence precision. Besides, it can also show whether the optimization method is trapped in local optima. According to Figure 3, NPSOSC achieves better fitness values than PSO in all the four datasets. NPSOSC has faster convergence speed than PSO. In the high-dimensional Colon dataset, the fitness value of NPSOSC in the 10th iteration is even better than the final fitness value of PSO. Benefiting from the two novel operators, NPSOSC can avoid local optimal solutions and further improve the fitness value afterwards. The convergence curves of the four datasets demonstrate the ability of NPSOSC in terms of convergence speed and locating optimal solution.

Figure 3

Convergence curves of NPSOSC and PSO in (a) Heart, (b) Ionosphere, (c) Musk1, and (d) Colon datasets



4.5. Analysis on the Neighborhood Search Strategy

This paper proposed a novel distance and fitness based neighborhood search strategy in which a weighting method is used to select the *lbest* for each particle. This strategy is able to form stable neighborhood structure and promote the exploitation in the promising areas. To prove the efficacy of the proposed neighborhood strategy, it is compared with the traditional neighborhood strategy which is described in Subsection 3.3. Table 7 compares classification accuracy and feature subset size of the two methods, in which NPSOSC1 represents the proposed algorithm and NPSOSC2 adopts the traditional neighborhood search strategy. NPSOSC1 achieves better classification accuracies than NPSOSC2 in 12 out of 16 datasets. In terms of the feature subset size, NPSOSC1 produces smaller feature subsets in 13 datasets. The results indicate that the distance and fitness based neighborhood search strategy is more successful than the traditional neighborhood search strategy in feature selection problems.

4.6. Discussions

According to the experimental results, the performance of the proposed NPSOSC can be summarized as follows:

1 NPSOSC outperforms other meta-heuristics in terms of classification accuracy and feature subset size on majority of the datasets.

2 The Wilcox sum test confirms the superiority of NPSOSC over other methods is significant on majority of the datasets.

3 The convergence curves show NPSOSC can escape from local optima and converge to (near) optimal solution quickly.

4 The new neighborhood search strategy can achieve higher classification accuracy and smaller feature subset size when compared with the traditional neighborhood strategy.

On a whole, the experimental results prove that the NPSOSC is a powerful wrapper based feature selection approach which is able to explore the entire search space more efficiently and preserve better population diversity. The reason for the superior performance of NPSOSC can be attributed to the novel neighborhood search strategy and the sine cosine mutation operator. These two improvements strengthen the search ability of the algorithm and lead to a better balance between exploration and exploitation.

5. Conclusion

In this work, a novel neighborhood based PSO with sine cosine mutation strategy is proposed for feature selection. PSO has fast convergence speed but it suffers from local optima stagnation and premature convergence. The goal of this study is to overcome the shortcomings of PSO to enhance its performance in feature selection problems. A distance and fitness based neighborhood search strategy is proposed to form stable and appropriate neighborhood structure for the particles. Each particle adopts local information to guide its search process instead of learning from the unique *gbest*. This search strategy can explore the entire feature space more efficiently and preserve better population diversity. A sine cosine mutation operator is introduced to PSO to strengthen its global exploration ability. Therefore, the proposed algorithm is able to achieve a better balance between global exploration and local exploitation. To validate the performance of the proposed NPSOSC, experiments are conducted on 16 datasets and seven state-of-the-art feature selection algorithms are used for comparison. Experimental results demonstrate that the proposed NPSOSC has the ability to effectively explore the entire search space and locate (near) optimal feature subsets. The statistical tests support the superiority of NPSOSC over other methods is significant. The analysis on the convergence process shows that NPSOSC can converge to optimal solutions quickly and escape from local optima.

For future work, the proposed NPSOSC can be extended to multi-objective feature selection model to optimize the classification accuracy and the feature subset size simultaneously. Another perspective is to develop hybrid feature selection model which is able to combine the advantages of both filters and wrappers.

Acknowledgement

This work was supported by the Research Foundation of Huangshan University (2021xkjq013).

References

1. Abdel-Basset, M., El-Shahat, D., El-Henawy, I. A New Fusion of Grey Wolf Optimizer Algorithm with a Two-Phase Mutation for Feature Selection. Expert Systems with Applications, 2019, 139, 112824. https://doi.org/10.1016/j.eswa.2019.112824
2. Ahmad, S., Mafarja, M., Faris, H., Aljarah, I. Feature Selection Using Salp Swarm Algorithm with Chaos. In: 2nd International Conference on Intelligent Sys-tems. 2018, 65-69. https://doi.org/10.1145/3206185.3206198
3. Aladeemy, M., Adwan, L. Booth, A., et al. New Fea-ture Selection Methods Based on Opposition Based Learning and Self-Adaptive Cohort Intelligence for Predicting Patient No-shows. Applied Soft Compu-ting, 2020, 86, 105866. https://doi.org/10.1016/j.asoc.2019.105866
4. Alazzam, H., Sharieh, A., Sabri, K. A Feature Selec-tion Algorithm for Intrusion Detection System Based on Pigeon Inspired Optimizer. Expert Systems with Applications. 2020, 148, 113249. https://doi.org/10.1016/j.eswa.2020.113249
5. Arora, S., Anand, P. Binary Butterfly Optimization Approaches for Feature Selection. Expert Systems with Applications, 2019, 116, 147160. https://doi.org/10.1016/j.eswa.2018.08.051
6. Asif, M., Nagra, A., Ahmad, M., Masood, K. Feature Selection Empowered by Self-Inertia Weight Adap-tive

Particle Swarm Optimization for Text Classifica-tion. Applied Artificial Intelligence, 2022, 36(1), 2004345. https://doi.org/10.1080/08839514.2021.2004345

7. Chen, K., Zhou, F., Yuan, X. Hybrid Particle Swarm Optimization with Spiral-Shaped Mechanism for Feature Selection. Expert Systems with Applications, 2019, 128, 140-156. https://doi.org/10.1016/j.eswa.2019.03.039
8. Chen, Z., Chen, Q., Zhang, Y. Clustering-based fea-ture Subset Selection with Analysis on the redundan-cy Complementarity Dimension. Computer Commu-nications, 2021, 168, 65-74. https://doi.org/10.1016/j.comcom.2021.01.005
9. Emary, E., Zawbaa, H. M., Hassanien, A. Binary Grey Wolf Optimization Approaches for Feature Se-lection. Neurocomputing, 2016, 172, 371-381.https://doi.org/10.1016/j.neucom.2015.06.083
10. Engelbrecht, A., Grobler, J., Langeveld, J. Set Based Particle Swarm Optimization for the Feature Selec-tion Problem. Engineering Applications of Artificial Intelligence, 2019, 85, 324-336. https://doi.org/10.1016/j.engappai.2019.06.008
11. Gu, S., Cheng, R., Jin, Y. Feature Selection for High-Dimensional Classification Using a Competitive Swarm Optimizer. Soft Computing, 2018, 22(3), 811-822. https://doi.org/10.1007/s00500-016-2385-6
12. Houssein, E., Gad, A., Hussain, K., Suganthan, P. Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application. Swarm and Evo-lutionary Computation, 2021, 63, 100868. https://doi.org/10.1016/j.swevo.2021.100868
13. Hu, G., Du, B., Wang, X., Wei, G. An Enhanced Black Widow Optimization Algorithm for Feature Se-lection. Knowledge-Based Systems, 2022, 235, 107638. https://doi.org/10.1016/j.knosys.2021.107638
14. Hu, P., Pan, J., Chua, S., Sun, C. Multi-surrogate Assisted Binary Particle Swarm Optimization Algorithm and Its Application for Feature Selection. Applied Soft Computing, 2022, 121, 108736. https://doi.org/10.1016/j.asoc.2022.108736
15. Kennedy, J., Eberhart, R.C., Particle Swarm Opti-mi-zation. In: Proceedings of IEEE International Con-fer-ence on Neural Networks, 1995, 1942-1948. https://doi.org/10.1109/ICNN.1995.488968
16. Kennedy, J., Eberhart, R.C. A Discrete Binary Ver-sion of the Particle Swarm Algorithm. In: Proceed-ings of IEEE Conference on Systems Man and Cyber-netics, 1997, 4104-4108. https://doi.org/10.1109/ICSMC.1997.637339
17. Köse, O., Oktay, T., Quadrotor Flight System Design Using Collective and Differential Morphing with SPSA and ANN. International Journal of Intelligent Systems and Applications in Engineering, 2021, 9(4), 159-164. https://doi.org/10.18201/ijisae.2021473634
18. Li, J., Cheng, K., Wang, S., Morstatter, F. Trevino, R. P.,Tang, J. L., Liu, H. Feature Selection: A Data Per-spective. ACM Computing Surveys, 2016, 50(6), 94. https://doi.org/10.1145/3136625
19. Li, T., Dong, H., Sun, J. Binary Differential Evolution Based on Individual Entropy for Feature Subset Op-timization. IEEE Access, 2019, 7, 24109-24121. https://doi.org/10.1109/ACCESS.2019.2900078
20. Mafarja, M., Aljarah, I., Faris, H., et al. Binary Grasshopper Optimisation Algorithm Approaches for Feature Selection Problems. Expert Systems with Applications, 2019, 117, 267-286. https://doi.org/10.1016/j.eswa.2018.09.015
21. Mafarja, M., Mirjalili, S. Whale Optimization Ap-proaches for Wrapper Feature Selection. Applied Soft Computing, 2018, 62, 441-453. https://doi.org/10.1016/j.asoc.2017.11.006
22. Maleki, N., Zeinali, Y., Niaki, S. A k-NN Method for Lung Cancer Prognosis with the Use of a Genetic Al-gorithm for Feature Selection. Expert Systems with Applications, 2021, 164, 113981. https://doi.org/10.1016/j.eswa.2020.113981
23. Mirjalili, S., Lewis, A. Grey wolf optimizer. Advances in Engineering Software, 2014, 69, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007
24. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. Knowledge-Based Systems, 2016, 96, 120-133. https://doi.org/10.1016/j.knosys.2015.12.022
25. Moradi, P., Gholampour, M. A Hybrid Particle Swarm Optimization for Feature Subset Selection by Integrating a Novel Local Search Strategy. Applied Soft Computing, 2016, 43, 117-130. https://doi.org/10.1016/j.asoc.2016.01.044
26. Neggaz, N., Houssein, E., Hussain, K. An Efficient Henry Gas Solubility Optimization for Feature Selec-tion. Expert Systems with Applications, 2020, 152, 113364. https://doi.org/10.1016/j.eswa.2020.113364
27. Nguyen, B., Xue, B., Zhang, M. A Survey on Swarm Intelligence Approaches to Feature Selection in Data Mining. Swarm and Evolutionary Computation, 2020, 54, 100663. https://doi.org/10.1016/j.swevo.2020.100663

28. Pan, L. Zhao, Y. Li, L. Neighborhood-based particle Swarm Optimization with Discrete Crossover for Nonlinear Equation Systems. Swarm and Evoluti-on-ary Computation, 2022, 69, 101019. https://doi.org/10.1016/j.swevo.2021.101019

29. Pervaizm, S., Haider, W., Ashraf, A., et al. Compara-tive Research Directions of Population Initialization Techniques Using PSO Algorithm. Intelligent Auto-mation & Soft Computing, 2022, 32 (3), 1427-1444. https://doi.org/10.32604/iasc.2022.017304

30. Qiu, C. A Hybrid Two-stage Feature Selection Me-th-od Based on Differential Evolution. Journal of Intel-li-gent & Fuzzy Systems, 2020, 39(3), 114. https://doi.org/10.3233/JIFS-191765

31. Qiu, C. A Novel Multi-swarm Particle Swarm Optimization for Feature Selection. Genetic Programming and Evolvable Machines, 2019, 20(4), 503-529. https://doi.org/10.1007/s10710-019-09358-0

32. Qiu, C. Bare Bones Particle Swarm Optimization with Adaptive Chaotic Jump for Feature Selection in Classification. International Journal of Computational Intelligence Systems, 2018, 11(1), 1-14. https://doi.org/10.2991/ijcis.11.1.1

33. Qiu, C., Liu, N. A Novel Three Layer Particle Swarm Optimization for Feature Selection. Journal of Intel-ligent & Fuzzy Systems, 2021, 41(1), 2469-2483. https://doi.org/10.3233/JIFS-202647

34. Qu, B., Suganthan, P. N., Das, S. A Distance-based Locally Informed Particle Swarm Model for Multi-modal Optimization. IEEE Transactions on Evolu-tionary Computation, 2012, 17(3), 387-402. https://doi.org/10.1016/j.swevo.2021.100868

35. Raymer, M., Punch, W., Goodman, E. Dimensionality Reduction Using Genetic Algorithms. IEEE Transac-ti-ons on Evolutionary Computation, 2000, 4 (2), 164-171. https://doi.org/10.1109/4235.850656

36. Tang, X., Dai, Y., Xiang, Y. Feature Selection Based on Feature Interactions with Application to Text Categorization. Expert Systems with Applications, 2019, 120, 207-216.https://doi.org/10.1016/j.eswa.2018.11.018

37. Thaher, T., Chantar, H., Too, J. Boolean Particle Swarm Optimization with Various Evolutionary Population Dynamics Approaches for Feature Selection Problem. Expert Systems with Applications, 2022, 195, 16550. https://doi.org/10.1016/j.eswa.2019.03.039

38. Tran, B., Xue, B., Zhang, M. Variable-length particle Swarm Optimization for Feature Selection on High-dimensional Classification. IEEE Transactions on Evolutionary Computation, 2019, 23(3), 473-487. https://doi.org/10.1109/TEVC.2018.2869405

39. Tran, B., Xue, B., Zhang, M. A New Representation in PSO for Discretization-based Feature Selection. IEEE Transactions on Cybernetics, 2017, 48, 1733-1746. https://doi.org/10.1109/TCYB.2017.2714145

40. Xie, Q., Cheng, G., Zhang, X., Lei, P. Feature Selec-tion Using Improved Forest Optimization Algorithm. Information Technology and Control, 2020, 49(1), 289-301. https://doi.org/10.5755/j01.itc.49.2.24858

41. Xue, B., Zhang, M., Browne, W.N. Particle Swarm Optimization for Feature Selection in Classification: Novel Initialization and Updating Mechanisms. Ap-plied Soft Computing, 2014, 18, 261-276. https://doi.org/10.1016/j.asoc.2013.09.018

42. Yan, X., Nazmi, S., Erol, B. An Efficient Unsuper-vised Feature Selection Procedure Through Feature Clustering. Pattern Recognition Letters. 2020, 131, 277-284. https://doi.org/10.1016/j.patrec.2019.12.022

43. Zhang, X., Xu, Y., Yu, C. A. Heidari and et al. Gaussian Mutational Chaotic Fruit Fly-built Optimization and Feature Selection. Expert Systems with Applica-tions, 2020, 141, 112976. https://doi.org/10.1016/j.eswa.2019.112976

44. Zorarpac, E., Ozel, S. A Hybrid Approach of Differ-ential Evolution and Artificial Bee Colony for Feature Selection. Expert Systems with Applications, 2016, 62, 91-103. https://doi.org/10.1016/j.eswa.2016.06.004