Intelligent Data Analysis 22 (2018) 191–210
DOI 10.3233/IDA-163392
IOS Press

Fault diagnosis method based on supervised particle swarm optimization classification algorithm

Bo Zheng[a,b], Hong-Zhong Huang[a,*], Wei Guo[a], Yan-Feng Li[a] and Jinhua Mi[a]
[a]*Center for System Reliability and Safety, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[b]*Civil Aviation Flight University of China, Guanghan, Sichuan 618307, China*

Abstract. A novel supervised particle swarm optimization (S-PSO) classification algorithm is proposed for fault diagnosis. In order to improve the accuracy of fault diagnosis and obtain the global optimal solutions with a higher probability, two strategies, i.e. a hybrid particle position updating strategy and a fixed iteration interval intervention updating strategy, are designed to balance the effect of the local and the global search. These methods increase the diversity of particles, expand the particles ability of searching the entire solution space, and guide the particles adaptively jumping out of the local optimal area. Meanwhile, based on the shorter intra-class distance, longer inter-class distance and maximum classification accuracy of training samples, a fitness function is designed to constraint the output optimal class centers. Experimental results demonstrate that the proposed S-PSO classification algorithm can overcome the problems in the classical clustering algorithms, which only consider the similarity of data instead of their physical meanings. The comparison on GE90 engine borescope image texture feature classification is also conducted. The results show that the performance of S-PSO classification algorithm is robust. Its classification accuracy is higher than those of popular methods, including support vector machine (SVM), neural network, Bayesian classifier, and k-nearest neighbor (k-NN) algorithm.

Keywords: Particle swarm optimization, classification algorithm, hybrid particle position updating strategy, fixed iteration interval intervention updating strategy, fitness function, fault diagnosis

1. Introduction

Due to the continual improvement of the technical level, analysis and judgment on the fault mechanisms and fault modes of modern equipments become more difficult. Meanwhile, the influence on the production activities caused by faults is increasingly serious. For some key parts, such as engine and spindle bearing, their healthy statuses have a direct and critical relationship with the safety of the production activities. The failure of critical equipments is the largest contributor to machine downtime, even catastrophic accidents in the automotive manufacturing industry [1]. In industry, it is unacceptable

*Corresponding author: Hong-Zhong Huang, Center for System Reliability and Safety, University of Electronic Science and Technology of China, No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chengdu, Sichuan 611731, China. Tel.: +86 28 6183 1252; Fax: +86 28 6183 0227; E-mail: hzhuang@uestc.edu.cn.

to face unscheduled breakdowns and production losses. So it is greatly significant to carry out the research on fault diagnosis, which can improve the efficiency of trouble-shoot, shorten the maintenance period, reduce the maintenance costs, and ensure production safety.

In fact, it is difficult to use the equipment operation features based on traditional analytical models to accurately describe the operation process of complex equipment, and thus their development is limited. However, with the rapid development of the intelligence technology, research on the large state feature data promotes further development of fault diagnosis technology, and intelligence technology focuses on the analysis and discrimination of the monitoring data without considering the internal mechanism of the monitored object. Therefore, intelligence technology has been widely applied in fault classification. Some classic intelligence methods, such as artificial neural network, statistical pattern recognition, and kernel-based algorithms, and so on, have been paid much attention and get rapid development. A large number of researches were proposed to promote the development of fault diagnosis technology in a more practical direction [2–5]. However, artificial neural network always has the drawback of poor convergence and generalization capability, and the network topology need to be determined artificially in advance [6]. Statistical pattern recognition is sensitive to the probability distribution of monitoring data [7]. Kernel-based algorithms are still confronted with the problems of optimizing the kernel parameters [8].

With the develop of bionic sciences, the swarm intelligent algorithms imitating natural phenomena have been widely used to solve various real-world problems including nonlinear process control, machine design, text mining, data clustering, features extraction, system optimization, grouping problems, path planning [9–12], and so on. The particle swarm optimization (PSO) algorithm is one of classic swarm intelligent algorithms. In the field of data analysis, PSO is mainly used as data clustering or as an auxiliary tool to optimize other classification algorithms. Cagnina et al. [13] improved the accuracy and efficiency of short text clustering by using a novel discrete PSO. Lam et al. [14] proposed a PSO-based K-Means algorithm for unsupervised gene clustering, and experimental results show that this proposed algorithm is more effective in reducing clustering error and improving convergence rate. Avanija and Ramar [15] proposed an ontology-based clustering algorithm using semantic similarity measure and PSO, and the experimental result shows that the proposed method is feasible and performs better than other traditional methods. Zhang et al. [16] introduced dynamical crossover with variable lengths and positions to PSO and proposed an improved method to solve the problem of K-Means and obtain correct clustering results. Fathi and Montazer [17] proposed a new learning method based on a novel PSO for radial basis function (RBF) to optimize the optimum steepest decent (OSD) algorithm, which improves the classification ability of RBF. Guraksin et al. [18] used PSO to generate a new training instance for SVM and obtained higher classification accuracy. Pattern recognition as the main fault diagnosis tool has been widely used to solve many real-world fault diagnosis problems in the machinery industry [19], in which clustering and classification are main techniques for pattern recognition. Obviously, in most of classification applications, the research on PSO is few, so it is worth researching on the classification ability of PSO deeper for expanding the fault diagnosis technical methods.

It is known that the PSO is based on the collaborative swarm heuristic search algorithm, which can provide solutions closer to the global optimal area on different engineering problems due to its concise mathematical expressions and effective search strategy. However, similar to other swarm intelligent algorithms, an open problem of the PSO is that it may trap into local suboptimal areas due to the premature convergence, which is caused by the lack of diversity, especially for complex multi-mode problems. Therefore, many modifications were proposed to solve this problem, such as parameter adjustment [20–22], neighborhood topology [23–25], and human behaviors [26–28], and so on. In this paper, new updating strategies are designed to obtain better optimization performance, which aims to overcome defects mentioned above.

Virtually, the essence of pattern recognition is to find the mapping between features and labels. In this way, we can get the unknown data label by the existing mapping. In this process, if an algorithm is trained by the data possessing labels, it can be defined as supervised algorithm, the process is called classification. Conversely, it is an unsupervised algorithm, and the process is called clustering. Meanwhile, the previous studies have discussed the principles and defects of clustering in detail [29–32]. It only relies on similar features among data and ignores the actual meaning of the data. Thus, there is often a big difference between the clustering results and true class in recognition, so clustering is not good at fault diagnosis. To solve this problem, a supervised PSO classification algorithm is proposed for classification of fault data collected from modern complex equipments. The proposed method is designed to overcome the shortcomings of normal clustering methods, and improve the accuracy of fault pattern recognition, so that faults can be located rapidly and accurately in the machinery industry.

The rest of the paper is organized as follows. In Section 2, the motivation and principle of the proposed S-PSO algorithm are introduced. Section 3 introduces the evolution process of PSO in detail. The application of engine is given in Section 4. Finally, in the last section conclusion is drawn.

2. Motivation and principle of S-PSO algorithm

2.1. Motivation of the prior class labels for classification

Because the operation characteristics, working environment, and load strength are not consistent, or there are different reasons and mechanisms for various faults, multi-dimensional monitoring data collected from monitored components exhibit the differences of equipment operations. It is necessary to use a signal processing method to analyze these monitoring data and then rapidly determine the state of the equipment operation, so that rational maintenance strategy can be made.

For discussing the classification method, the prior class labels are defined: the monitoring data, which reflect the same state or fault type of the equipment, can be classified into a category, and the prior label, $label_i(i = 1, \cdots, n_c)$ for class i is then assigned, where n_c is the total number of classes. These prior class labels are the supervised signals for training the classification method. Only if we correctly and reasonably assign labels for known data will we be able to classify unknown data accurately and rapidly. To finish this task, some assumptions need to be emphasized:

(1) Although the data in the same class can reflect the same state of equipment, the distribution properties of data present certain distribution shape in the multi-dimensional space, such as reunion distribution and strip distribution, even irregular distribution, which proves the data in the same class also have differences; moreover, there is no obvious boundary between the data in different classes. In this case, the clustering always causes the recognition error for training data easily, and the prior class label can accurately identify these data in the same class, and prevent the error of classifying unknown data. Based on these analyses, Zheng and Gao [32] compared the recognition performance between clustering and classification, and the results show the classification is better.

(2) The recognition process of clustering needs to predefine class labels of training data, but the training data in classification are given labels in advance. Consequently, classification helps to classify unknown data rapidly.

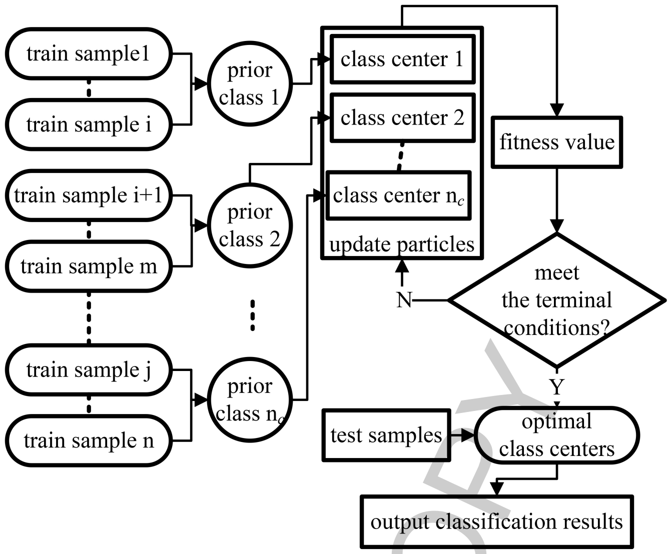

Fig. 1. S-PSO classification schematic diagram.

If there is no prior class label, every training sample needs to be set a class label randomly, and then the classification will degenerate into the clustering process. The main problem of the clustering algorithm lying in the above assumptions may reduce the recognition accuracy. Therefore, the proper prior class labels have positive effect on the guidance of the classification.

2.2. Principle of the S-PSO classification algorithm

The schematic diagram of the S-PSO classification algorithm is shown in Fig. 1.

Actually, many definitions of distance have been used to measure the similar relationship of vectors, such as Minkowski distance, Mahalanobis distance, Manhattan distance, Chebyshev distance, Pearson correlation coefficient, and Cosine similarity [33–36]. These definitions measure the similarity according to different theories. Generally, Euclidean distance has been widely used, and several popular algorithms are computed by Euclidean distance [29,32,37]. Euclidean distance is in the interval $[0, +\infty)$, where "0" represents two vectors are identical, that is to say, if the Euclidean distance of two vector is infinite, the similarity will be 0. In this paper, Euclidean distance is used due to its generality and intuition.

Let $\boldsymbol{x}_l = [x_{l1}, x_{l2}, \cdots, x_{ld}]\ (l = 1, \cdots, n)$ denotes the training sample, where n is the total number of training samples, d is the vector dimensions of training sample. The class center is $\boldsymbol{c}_i = [c_{i1}, c_{i2}, \cdots, c_{id}]\ (i = 1, \cdots, n_c)$ of different classes, and the distance between $\boldsymbol{x}_l$ and $\boldsymbol{c}_i$ is calculated by using Euclidean distance defined as follows:

$$dist_{li} = \sqrt{\sum_{j=1}^{d} (x_{lj} - c_{ij})^2}. \tag{1}$$

The distance from a certain training sample to its class center should be shortest. Determining all class centers that satisfy the requirement of training sample classification is a process of searching for the optimal solution depending on some criterion in $n_c \times d$ dimensional solution space, so different kinds of class centers as the particles' positions can be obtained by randomly searching. There are $n_c \times d$

variables to be optimized, and the criterion guiding search updating is the fitness function, fit$(\cdot)$. Once the terminal conditions of iteration are satisfied, all optimal class centers $\boldsymbol{c}_{(best)i}(i = 1, \cdots, n_c)$ will be obtained. The class for the test sample $\boldsymbol{y}_j$ is calculated as follows:

$$\text{S-PSO}(\boldsymbol{y}_j) = \langle i|dist_{ji} = \min(dist_{j1}, \cdots, dist_{jn_c}), i \in [1, \cdots, n_c]\rangle. \tag{2}$$

In essence, $\boldsymbol{y}_j$ belonging to that class i means the distance from $\boldsymbol{y}_j$ to $\boldsymbol{c}_{(best)i}$ is shortest.

The class labels in clustering are set randomly. After updating the particles, all training samples need to be re-clustered according to a certain criterion. The class centers are calculated again, and fitness values are then calculated using fitness function. As for the problem of non-identifiability, Zheng et al. [29] analyzed the possible solution for recognizing unknown samples in clustering. Therefore, the S-PSO classification algorithm is proposed in this paper to overcome the problems existed in normal clustering algorithms and improve the accuracy of fault classification.

3. S-PSO classification algorithm

3.1. Fitness function

Fitness function, known as the important theoretical foundation of PSO algorithm, directly influences the optimization effect of class centers. The clustering algorithm in [38] used the shorter intra-class distance from training samples to their own clustering center as its fitness function. Based on the work in [38], Nanda and Panda [39] added the longer inter-class distance to the fitness function. In a supervised classification algorithm, a supervised signal should be fully used to design the fitness function. Therefore, the classification accuracy of training samples is introduced into the fitness function, which is beneficial to improve the accuracy. We assume that we confront a maximization problem in this paper. The fitness function based on the shorter intra-class distance is formulated as follows:

$$\text{fit}_1(\boldsymbol{P}) = \frac{1}{\sum_{i=1}^{n_c}\sum_{l=1}^{ns_i}\sqrt{\sum_{j=1}^{d}(x_{lj} - c_{ij})^2}}, \tag{3}$$

The fitness function based on the shorter intra-class distance and longer inter-class distance in this paper is defined as follows:

$$\text{fit}_2(\boldsymbol{P}) = \frac{1}{\sum_{i=1}^{n_c}\sum_{l=1}^{ns_i}\sqrt{\sum_{d=1}^{D}(x_{ld} - c_{id})^2} - \sum_{i=1}^{n_c}\sum_{j=i}^{n_c}\sqrt{\sum_{d=1}^{D}(c_{jd} - c_{id})^2}}, \tag{4}$$

where ns_i is the number of training samples belonging to the certain class label i, and $\boldsymbol{P}$ is a particle. A larger value of fit$_2(\boldsymbol{P})$ means the shorter intra-class distance and the longer inter-class distance can ensure smaller errors on training sample classification.

The supervised signal used to express the classification accuracy of training samples can be used to find the optimal class centers. The classification accuracy is calculated by Eq. (5):

$$ac = \frac{n_r}{n}, \tag{5}$$

where n_r is the number of training samples classified into the correct class by Eq. (2). The modified fitness function is defined as follows:

$$\mathrm{fit}(\boldsymbol{P}) = \frac{1}{\sum_{i=1}^{n_c}\sum_{l=1}^{ns_i}\sqrt{\sum_{d=1}^{D}(x_{ld}-c_{id})^2} - \sum_{i=1}^{n_c}\sum_{j=i}^{n_c}\sqrt{\sum_{d=1}^{D}(c_{jd}-c_{id})^2}} + \frac{n_r}{n}. \tag{6}$$

The above fitness function, considering 3 factors of intra-class, inter-class distance, and classification accuracy on training samples, provides an accurate and clear guidance for S-PSO classification algorithm. The better value the $\mathrm{fit}(\boldsymbol{P})$ has, the easier the optimal class centers meet these three factors. It helps to improve the classification accuracy of test samples.

3.2. A novel particle position updating strategy

The traditional position updating principle of a traditional PSO algorithm can be found in [40]. The traditional algorithm is good at global search, which has the problem on obtaining the local optimum by tracing individual extremum and global extremum. To solve this, a differentiation random updating novel strategy enlightened by the intermediate value theorem is then proposed in this paper to enhance local search ability of particles.

3.2.1. A differentiation random positions updating strategy

According to intermediate value theorem, in a continuous search space, we can assume that the global optimal solution $\boldsymbol{P}_g$ is unknown, and $\boldsymbol{P}$ is an ordinary particle. Obviously, $\mathrm{fit}(\boldsymbol{P}) < \mathrm{fit}(\boldsymbol{P}_g)$, there must be a particle $\boldsymbol{P'}$ existed in the neighborhood taking the distance from $\boldsymbol{P}$ to $\boldsymbol{P}_g$ as radius and $\boldsymbol{P}$ as the center of hypersphere, so that $\mathrm{fit}(\boldsymbol{P}) < \mathrm{fit}(\boldsymbol{P'}) < \mathrm{fit}(\boldsymbol{P}_g)$ is tenable. Accordingly, a much better solution maybe exits in the neighborhood around a current solution. By means of this theorem, it is possible to gradually approximate the global optimal solution. Based on this, the particle with better fitness value should search a smaller solution space. In other words, the better value the $\mathrm{fit}(\boldsymbol{P})$ has, the faster speed the search has, and vice versa. Meanwhile, in order to maintain the diversity of particle swarm and increase convergence speed relatively, the moving direction of particles in each dimension is set randomly. The procedure of the differentiation random positions updating strategy is described as follows:

Step 1: The training and test samples should be normalized in the interval [0, 1]. It helps to eliminate influence of dimension, reduce the search area and improve the algorithm efficiency.

Step 2: Initialize particle position in the whole solution space. Assume that n_i is the number of training samples belonging to class i, and the initial class center $\boldsymbol{c}_i(i = 1, \cdots, n_c)$ is calculated by Eq. (7):

$$\begin{cases} \boldsymbol{c}_i = [c_{i1}, c_{i2}, \cdots, c_{ik}, \cdots, c_{id}] \\ c_{ik} = \mathrm{unifrnd}(\min(x_{1k}, x_{2k}, \cdots, x_{n_ik}), \max(x_{1k}, x_{2k}, \cdots, x_{n_ik}), 1, 1) \end{cases}, \tag{7}$$

where function $\mathrm{unifrnd}(\alpha, \beta, a, b)$ is used to generate uniform distribution real matrix with a rows and b columns on interval $[\alpha, \beta]$.

Step 3: Rank the particles in ascending order according to their fitness values.

Step 4: Generate the moving direction of particles in each dimension. Function $\mathrm{unifrnd}(-1, 1, n_c, d)$ is used to generate moving direction matrix $\boldsymbol{A}$ with n_c rows and d columns on interval [−1, 1]. Besides, if the element $a \geqslant 0$, $a = 1$, which means that a certain dimension of the particle moves along the positive direction of current position. If the element $a < 0$, $a = -1$, which means that a certain dimension of the particle moves along the negative direction of the current position.

Step 5: Calculate the search speeds of particles. In theory, each dimension of the particle should search the solution on maximum interval [0, 1]. Assume that $\text{fit}(\boldsymbol{P}_1) < \cdots < \text{fit}(\boldsymbol{P}_i) \cdots < \text{fit}(\boldsymbol{P}_{N_{\max}})$, where $N_{\max}$ is the total number of particles, and $i \in (1, \cdots, N_{\max})$. Obviously, $\boldsymbol{P}_{N_{\max}}$ need a smallest search speed, and the search speed of others will increase gradually. The search speed matrix of the *i-th* particle at certain iteration can be calculated as follows:

$$\boldsymbol{SP}_i^{(iter)} = \boldsymbol{A} \cdot \boldsymbol{D} \cdot \left(0.2 \cdot e^{-\frac{2\cdot(iter-1)}{(iter_{\max}-1)}} \cdot \left[1 + 10 \cdot \frac{(\text{fit}(\boldsymbol{P}_{N_{\max}}) - \text{fit}(\boldsymbol{P}_i))}{\text{fit}(\boldsymbol{P}_{N_{\max}})}\right]\right), \tag{8}$$

where $i \in [1, \cdots, N_{\max}]$ denotes the i^{th} particle, *iter* denotes the current iteration ordinal, $iter_{\max}$ is the maximum iteration number, and $\boldsymbol{D}$ is designed as disturbance matrix on interval [0, 2], which is a uniform distribution $n_c \times d$ matrix generated randomly. If $\boldsymbol{D}$ does not exist in Eq. (8), the values of elements in search speed matrix will be completely the same except their directions; regrettably, it is not conducive to enhance the diversity of the population. The reason of setting elements on interval [0, 2] is that the probabilities of values being greater or less than 1 are basically the same. Furthermore, with the increasing of iterations, the current optimal solution will be closer to the global optimal solution; thus, the search speeds of particles should be reduced. Of course, this situation has been taken into account in Eq. (8).

Step 6: Update the particles using the following equation:

$$\boldsymbol{P}_i^{(iter+1)} = \boldsymbol{P}_i^{(iter)} + \boldsymbol{SP}_i^{(iter)}, \tag{9}$$

Meanwhile, update the particles by means of elitism approach. If the $\text{fit}(\boldsymbol{P}_i^{(iter+1)}) > \text{fit}(\boldsymbol{P}_i^{(iter)})$, let $\boldsymbol{P}_g = \boldsymbol{P}_i^{(iter+1)}$, otherwise, let $\boldsymbol{P}_g = \boldsymbol{P}_i^{(iter)}$. Finally, store the global optimal $\boldsymbol{P}_g$.

Step 7: Stop the iteration until the terminal conditions are satisfied, or return to Step 3. The terminal conditions may include $ac = 100\%$, $iter = iter_{\max}$ or $\text{fit}(\boldsymbol{P}_g) = V_f$, where V_f is defined as the objective fitness value. An appropriate V_f can help to classify the test samples under the condition of reducing the iterative time. The larger value of V_f will make itself loss the significance as terminal conditions, while the smaller value is beneficial to reduce the iterative time, but isn't useful to accurately classify test samples, the concrete value of V_f relies on the separability in space of training samples. Thereafter, the purpose of defining V_f is to reduce the iterative time and further meet the classification requirements.

As mentioned above, the proposed updating strategy without tracking individual extremum and global extremum has great improvement on local search, and it can obtain global optimal solutions easily. However, it is not beneficial to finding the general situation of the solutions distribution, which means this strategy is not good at the global search. If the initial positions of the particles are close to the optimal solutions, it is easy to obtain optimal solutions; otherwise, it may obtain poor solutions. Therefore, this strategy is sensitive to the initial positions of the particles, which will affect the convergence rate. A hybrid updating strategy is then proposed to combine the traditional updating strategy and makes a balance between the global search and local search.

3.2.2. *A hybrid particle position updating strategy*

The schematic diagram of the proposed hybrid updating strategy is shown in Fig. 2. Before updating the particle positions, the particles need to be ranked in an ascending order according to their fitness values. The procedure of updating is described as follows:

(1) From particles 1 to m, the traditional updating strategy is conducted. In this step, the independent global extremum $\boldsymbol{P}_{g1}$ and the individual extremum $\boldsymbol{P}_{id}$ are calculated, and these particles only trace $\boldsymbol{P}_{g1}$ and $\boldsymbol{P}_{id}$.

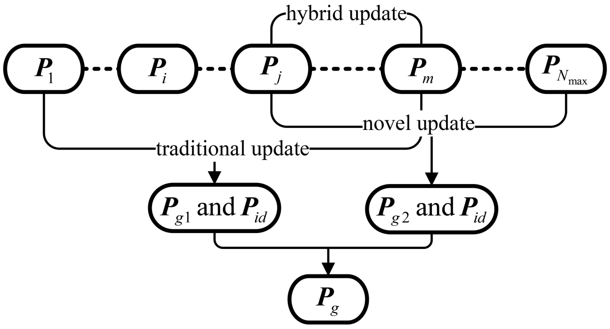

Fig. 2. Hybrid updating schematic diagram.

(2) The novel updating strategy mentioned above is conducted on the particles from j to $N_{\max}$. Similarly, the independent global extremum $\boldsymbol{P}_{g2}$ and the individual extremum $\boldsymbol{P}_{id}$ are also calculated.
(3) If $\text{fit}(\boldsymbol{P}_{g1}) < \text{fit}(\boldsymbol{P}_{g2})$, the global optimal $\boldsymbol{P}_g = \boldsymbol{P}_{g2}$; otherwise, $\boldsymbol{P}_g = \boldsymbol{P}_{g1}$. In this paper, m and j are presetting values.

m particles are updated by traditional strategy, which need the displacement velocities and inertia weights, and their initial velocities are generated by Eq. (10) because each dimension of the solution is on interval of [0, 1].

$$\boldsymbol{V}_i = \text{unifrnd}(-0.5, 0.5, n_c, d). \tag{10}$$

The element of $\boldsymbol{V}_i$ is set on interval $[-0.5, 0.5]$ initially, and the initial weight used in the traditional strategy is adjusted dynamically as the iteration proceeding and calculated as follows:

$$\omega^{(iter)} = \omega_s - iter \cdot (\omega_s - \omega_e)/iter_{\max}, \tag{11}$$

where $i \in [1, \cdots, N_{\max}]$, ω_s is the initial value of ω, and ω_e is the final value of ω. Usually, their values are $\omega_s = 0.9$ and $\omega_e = 0.2$. At the early stage of iteration, a larger value of ω helps to expand the solution space and increase the diversity of particles. In the later stage of iteration, a smaller value helps to converge fast and improve the capability of local search.

In essence, the advantage of the hybrid strategy lies in the mutual promotion of the traditional strategy and the novel strategy. Most of particles are updated by the traditional strategy and help to expand the solution space and increase the diversity, which can transport the excellent particles for the proposed hybrid strategy. Only a small part of excellent particles is updated by the hybrid strategy that has strong local search ability and provides more excellent particles than the traditional strategy. Furthermore, the hybrid strategy has more efficient updating, and accordingly it can obtain the optimal solution easily and rapidly.

3.3. The particles fixed iteration intervention updating strategy

Randomness, like a double-edge sword, has both positive and negative effects. On one hand, it can help to increase the diversity of the particles, and reduce the complicated mathematic calculation. On the other hand, it also increases the uncertainty in the optimal solutions. The influence of randomness may make final solutions fluctuating in the solution space, and may be non-uniqueness for each calculation, especially for complex problems. Therefore, the randomness may lead to the global optimal solution with a certain probability. One of solution is to use the intervention updating strategy, which can make some particles adaptively jump out of current search area under certain conditions and search the entire

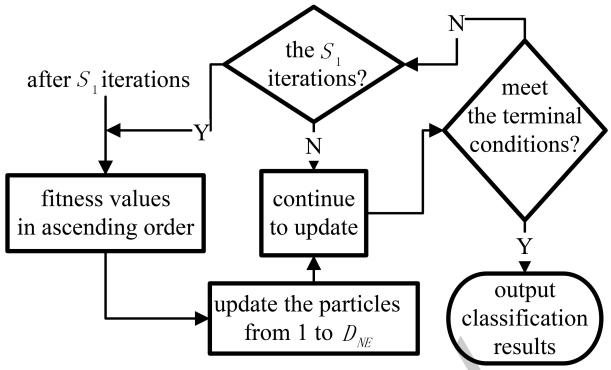

Fig. 3. Flow chart of the particles intervention updating strategy.

solution space as widely as possible. This strategy can ensure to obtain the global optimal solution with higher probability, reduce the influence of the randomness, overcome the defects of trapping into the local optimal area, and increase the diversity. In this paper, a fixed iteration interval intervention updating strategy based on dynamic neighborhood is thus proposed.

The schematic diagram of this strategy is shown in Fig. 3. Within the maximal iteration, a certain number of particles are firstly updated by Eq. (9) every S_1 iterations, where S_1 is defined as maturity period, without enough iterative process for these particles just updated, and then this intervention updating is nonsense, so the function of S_1 is to make the particles have proper iterative steps to converge to a certain optimal area. Before updating, these particles need to be ranked in an ascending order. Assume that $\text{fit}(\boldsymbol{P}_1) < \cdots < \text{fit}(\boldsymbol{P}_i) \cdots < \text{fit}(\boldsymbol{P}_{N_{\max}})$, where $i \in (1, \cdots, N_{\max})$, and then update the particles from 1 to D_{NE}, where D_{NE} is defined as dynamic neighborhood and calculated by Eq. (12):

$$\begin{cases} D_{NE} = \text{round}\left(N_{\max} \cdot \alpha^{(iter)}\right) \\ \alpha^{(iter)} = (\alpha_e - \alpha_s) \cdot (iter/iter_{\max}) + \alpha_s \end{cases}, \tag{12}$$

where $\text{round}(\cdot)$ is the round off function, α is defined as the dynamic change factor, α_s is initial value of α, and α_e is final value of α. Usually, $\alpha_s = 0.3$, and $\alpha_e = 0.8$. For each S_1 iteration, repeatedly update these particles according to their fitness values until the terminal conditions are satisfied.

The ability of PSO global optimization enhanced by this intervention updating strategy can be proven as follows:

Let $\boldsymbol{S}_{n_c \times d}$ denotes the solution space of all possible solutions about the optimal problem, $\boldsymbol{S}_{n_c \times d}$ is a subset of real space $\boldsymbol{R}_{n_c \times d}$, that is $\boldsymbol{S}_{n_c \times d} \subset \boldsymbol{R}_{n_c \times d}$, then the equation $P\left(\boldsymbol{P}_g \in \boldsymbol{S}_{n_c \times d}\right) = 1$ is tenable, which means the probability of the global optimal solution $\boldsymbol{P}_g$ belonging to solution space $\boldsymbol{S}_{n_c \times D}$ is 1. Every S_1 iteration, the particles from 1 to D_{NE} are updated. We assume that these particles are updated at the *iter* iteration. Before updating, the particles are ranked in ab ascending order, that is, $\text{fit}(\boldsymbol{P}_1) < \cdots < \text{fit}(\boldsymbol{P}_{D_{NE}}) \cdots < \text{fit}(\boldsymbol{P}_{N_{\max}})$, and $\boldsymbol{S}_{1n_c \times d}$ denotes the solution space enclosed by these particles. It is obvious that, after *iter* iterations, these particles have converged to a certain optimal area, which means $\boldsymbol{S}_{1n_c \times d} \subset \boldsymbol{S}_{n_c \times d}$. After that, according to the intervention updating strategy, the particles from 1 to D_{NE} are updated. In essence, updating strategy can extend these D_{NE} particles to the current solution space $\boldsymbol{S}_{1n_c \times d}$ and form a new solution space $\boldsymbol{S}_{2n_c \times d}$, so we can infer $\boldsymbol{S}_{1n_c \times d} \subseteq \boldsymbol{S}_{2n_c \times d} \subset \boldsymbol{S}_{n_c \times d}$, which means $P\left(\boldsymbol{P}_g \in \boldsymbol{S}_{1n_c \times d}\right) \leqslant P\left(\boldsymbol{P}_g \in \boldsymbol{S}_{2n_c \times d}\right)$. The particles finally converge from the new solution space $\boldsymbol{S}_{2n_c \times d}$. After the maturity period S_1, the

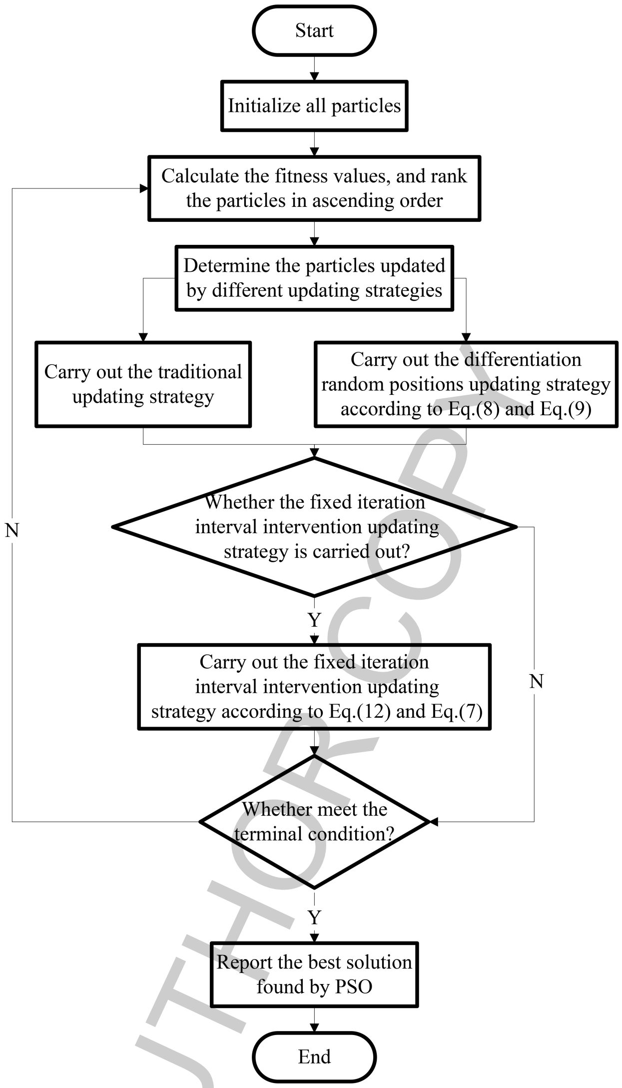

Fig. 4. Flowchart of the proposed PSO.

probability of obtaining the global optimal solution will increase. After several updating, the algorithm has higher probability to obtain the global optimal solution. The inference process indicates that within the $iter_{\max}$ iterations, the fixed iteration interval intervention updating strategy can increase the diversity and make the particles search in the entire solution space as much as possible, and ensure that the global optimal solution can be obtained with higher probability.

After using this strategy, at the early stage of iterations, updating a small proportion of particles with smaller fitness values not only increases the diversity of particles, but also keeps an acceptable convergence rate. At the later stage of iterations, updating a larger proportion of particles can help to increase the chance of jumping out of local optimal area. Therefore, this intervention updating strategy has a good balance between the efficiency of the algorithm and its optimization results. The flow chart of the proposed PSO is illustrated in Fig. 4.

Table 1
The relative informations about data sets

Name	Dimensions	Categories	Training samples	Test samples	Area
Seeds	7	3	120 (40 + 40 + 40)	80 (30 + 30 + 30)	Physical
Haberman's survival (HS)	3	2	121 (80 + 41)	105 (65 + 40)	Life
Wisconsin diagnostic breast cancer (WDBC)	9	2	503 (324 + 179)	180 (120 + 60)	Life
Indian liver patients (ILP)	10	2	213 (106 + 107)	120 (60 + 60)	Life

4. Experiments and analyses

4.1. Case 1: Data from UCI (University of California Irvine) database

This case is used to evaluate classification ability of algorithm using the commonly used data. Several typical data in the UCI database are used, and their relative information is shown in Table 1. Training and test samples are divided randomly. In this study, some widely used fault classifiers, such as SVM [41], Back Propagation (BP) network [42], Learning Vector Quantization (LVQ) network [43], the Bayes classifier based on expectation maximization (EM) and mixture normal distribution [44], and k-Nearest Neighborhood (k-NN) [45], will be used to compare with the S-PSO algorithm.

The PSO is a heuristic swarm random search algorithm and randomness exists in many aspects, i.e., random initial positions and velocities of the particles. The position updating and the intervention updating strategy also involve the randomness. Therefore, the randomness is the main factor affecting the performance of the PSO algorithm, which may obtain a local extremum. Meanwhile, it is well known that the influence of random weight setting in BP and LVQ may lead to the uncertainty of results; in addition, kernel parameters of SVM optimized by the method in [41] and the estimator of parameters based EM are still keeping uncertainly. To obtain a statistical soundness on the results and verify the ability of classification, we did 50 experiments continuously under the same conditions to get the statistical parameters of classification accuracies including Min, Mean, Max, and STD (Standard Deviation). k-NN is sensitive to the parameter k, so we will verify the influence of difference values of k from 1 to the number of training samples. Parameters for the S-PSO are set as follows: the total number of particles $N_{\max} = 60$, the maximum iteration number $iter_{\max} = 400$, and maturity period $S_1 = 100$. The particles from 1 to 58 are updated by the traditional updating strategy, and the particles from 54 to 60 are updated by the differentiation random positions updating strategy. The terminal condition is $iter = iter_{\max}$. For more details about the settings of SVM, BP and LVQ network, please refer to references cited above. The error goals for BP and LVQ network are set as 0.001, and their iteration numbers are set to 1000 and 200, respectively. The number of mixture normal distribution is set to three. The error goal is 10^{-6} and the iteration numbers of EM is set to 3000.

Actually, although these algorithms are all used to classify data, each of them has its own principles. SVM relies on the nonlinear hyper-plane; BP network relies on the nonlinear mapping relation; LVQ network relies on the distance to the winner neurons; Bayes takes the mixture normal distribution of training samples as prior probability distribution; and k-NN needs to determine the number of category of k nearest neighbors that also depends on the distances. As shown in Table 2, the classification performance of the proposed S-PSO is better than other algorithms. It indicates that the S-PSO classification algorithm has good classification ability to different dimensions of data, and randomness has made little or even no effect on S-PSO algorithm. Obviously, it can avoid trapping into a local optimal area and realize the accurate classification to the commonly used data.

Table 2
Classification accuracies comparison of S-PSO and other algorithms when using the 4 data sets

Data set	Algorithm	Min	Mean	Max	STD
Seeds	S-PSO	0.9556	0.9685	0.9778	0.0078
	SVM	0.9556	0.9556	0.9556	0
	BP	0.9333	0.9571	0.9778	0.0110
	LVQ	0.9444	0.9656	0.9778	0.0088
	Bayes	0.9333	0.9484	0.9667	0.0118
	k-NN	0.9222 ($k = 87$)	0.9456	0.9778 ($k = 4$)	0.0120
HS	S-PSO	0.6667	0.6857	0.6952	0.0076
	SVM	0.3619	0.4426	0.4952	0.0517
	BP	0.3238	0.5008	0.7048	0.1096
	LVQ	0.4381	0.5862	0.6857	0.0736
	Bayes	0.3333	0.5121	0.7046	0.1162
	k-NN	0.5000 ($k = 3$)	0.6160	0.6698 ($k = 10$)	0.0248
WDBC	S-PSO	0.9611	0.9611	0.9611	0
	SVM	0.9611	0.9611	0.9611	0
	BP	0.8667	0.9214	0.9500	0.0157
	LVQ	0.9000	0.9181	0.9389	0.0114
	Bayes	0.9444	0.9464	0.9611	0.0045
	k-NN	0.6667 ($k = 359$)	0.8551	0.9667 ($k = 7$)	0.1248
ILP	S-PSO	0.7167	0.7379	0.7750	0.0170
	SVM	0.425	0.4393	0.4583	0.0106
	BP	0.2667	0.4165	0.5417	0.0621
	LVQ	0.5417	0.6700	0.7250	0.0451
	Bayes	0.4333	0.6051	0.7000	0.0531
	k-NN	0.4750 ($k = 3$)	0.5949	0.6750 ($k = 34$)	0.0447

4.2. Case 2: Damage classification of an aeroengine

Aeroengine is a great achievement in the modern machinery industry. It is a complicated, huge and critical electronic and mechanical integrated system. Under the constantly working condition of high temperature, high pressure, and strong vibration, main mechanical components, such as blades, disk, spindle, have to suffer from corrosion, wear and fatigue. Serious faults, including performance degradation, abnormal vibration, and severe abrasion, are caused by these mechanical components. It has been proven that aeroengine's malfunction contributes main threats to the flight safety [42]. The borescope image is a commonly used tool to detect mechanical component damages of the crackle, including tearing, corrosion, curling, burn, groove, and so on. These damages are usually caused by overheated internal environment, vibration, wear, erosion, and strike. Because the working environment of each mechanical component is different, these damages often occur on a specific mechanical component [47]. By using the Content-Based Image Retrieval (CBIR), the features, including color, texture, and shape from borescope images can be extracted, and then the database about the damage images can be constructed. Using this database, the diagnosis can be realized. In reality, the diagnosis results are valuable reference to the engineers and is also possible to improve the efficiency of fault diagnosis. Moreover, the automatic diagnostic results and the engineer's judgments can confirm each others, which decreases the misjudgment rate and ensure the aeroengine operation reliability and safety.

Using the CBIR, Tang [48] firstly extracted mean and covariance of the surface texture parameters, including angular second moment, contrast, correlation, covariance, and inverse difference moment from the gray level co-occurrence matrixes calculated from the borescope images of GE90 aeroengine, and then constructed the database with 10 features of four borescope damage images, i.e. blade tip curling,

Table 3
GE90 engine borescope image texture features

No.	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	*label*
1	0.1337	0.0064	0.1461	0.0482	0.9841	0.0052	1.0928	0.0394	0.9484	0.0149	1
						⋮					
54	0.1478	0.0073	0.1429	0.0478	0.9819	0.0060	1.0608	0.0410	0.9453	0.0160	4
1	0.1043	0.0022	0.0624	0.0176	0.9880	0.0005	1.1708	0.0240	0.9694	0.0084	1
						⋮					
26	0.0979	0.0022	0.0617	0.0175	0.9975	0.0006	1.1590	0.0236	0.9708	0.0080	4

a (blade tip curling) b (corrosion) c (crackle) d (tearing)

Fig. 5. Four damage images of GE90 engine.

corrosion, crackle, and tearing, which are shown in Fig. 5. Eighty samples were applied to the proposed method, in which 54 samples are used as training samples, and the rest are test samples. Part of samples are shown in Table 3, in which the meanings of headers are: f_1 – the mean of angular second moment, f_2 – the covariance of angular second moment, f_3 – denotes the mean of contrast, f_4 – the covariance of contrast, f_5 – the mean of correlation, f_6 – the covariance of correlation, f_7 – the mean of covariance, f_8 – the covariance of covariance, f_9 – the mean of inverse difference moment, f_{10} – the covariance of inverse difference moment; the prior class label *label* – damage types, and the numbers 1, 2, 3, and 4 of *label* represent blade tip curling, corrosion, crackle, and tearing, respectively.

In Table 3, there are 54 samples for training. The first damage has 10 samples (Sample 1 $\sim$ 10). The second damage has 12 samples (Sample 11 $\sim$ 22). The third damage has 8 samples (Sample 23 $\sim$ 30). The fourth damage has 24 samples (Sample 31 $\sim$ 54). Twenty-six samples are used for test, in which the first damage has 6 samples (Sample 1 $\sim$ 6), the second damage has 6 samples (Sample 7 $\sim$ 12), the third damage has 6 samples (Sample 13 $\sim$ 18), and the fourth damage has 8 samples (Sample 19 $\sim$ 26).

4.2.1. *Experimental results and analyses*

To demonstrate the superiority of the proposed classification algorithm in fault diagnosis, common K-Means, fuzzy C-Means, the improved Shuffled Frog Leaping Algorithm (SFLA), and PSO clustering algorithm [9,10] are also applied to diagnose the four damage types of GE90 engine for the purpose of comparing with the S-PSO classification algorithm. The specific discussion about SFLA can be found in Refs. [49,50]. For this diagnosis problem, the population size of SFLA are 50 frogs that are divided into 5 memeplexes. The iteration number for each memeplex is 200 for the local search, and 200 iterations process of all memeplexes repeats 10 times. Thus, 5 memplexes, 200 iterations for each memplex, and 10 times for total iteration are parameters for the SFLA. Fifty particles are used in PSO algorithm, and the maximum iteration number $iter_{\max} = 600$. The results of these clustering algorithms are shown in Table 4. The numbers in this table correspond to the sequence numbers of the training. As shown in

Table 4
Comparison of clustering results in Case 2

	Class 1	Class 2	Class 3	Class 4
True state	1–10	11–22	23–30	31–54
K-Means	1–10, 26, 35, 43, 51	11–22, 24, 28, 32, 33, 40, 41, 48, 49	23, 25, 27, 29,30	31, 34, 36–39, 42, 44–47, 50, 52–54
Fuzzy C-Means	7–10, 27, 28, 31–36, 43, 51	11–16, 18, 29, 30	17, 19–23, 25	1–6, 24, 26, 37–40, 50, 52–54
PSO cluster	1–10, 26, 35, 43, 51	11–22, 24, 28, 32, 33, 40, 41, 48, 49	23, 25, 27, 29, 30	31, 34, 36–39, 42, 44–47, 50, 52–54
SFLA cluster	1–10, 26, 35, 43, 51	11, 13–22, 24, 28	23, 25, 27, 29, 30	12, 31–34, 36–42, 44–50, 52–54

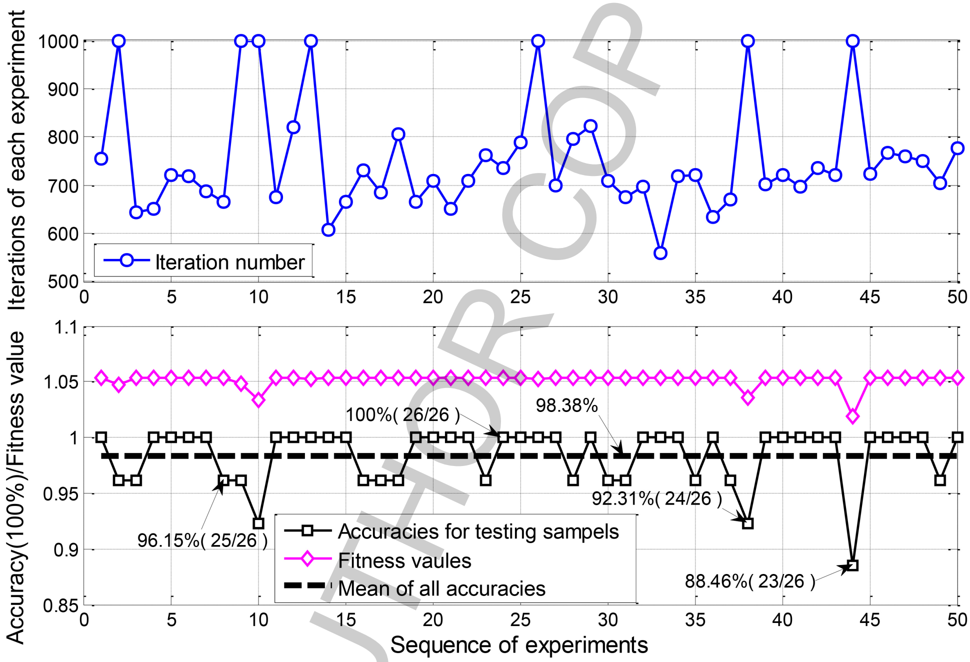

Fig. 6. Comparison of 50 experiments.

Table 4, clustering results calculated by SFLA are relatively close to real states, while the others have obvious differences. The reason is that these clustering algorithms only consider the similarity of distance, correlation, and so on, reflected by the training samples, don't consider the practical physical meanings contained in the samples. Therefore, it is impossible for these clustering algorithms to guide the classification of the test samples commendably. Based on the analysis above, the S-PSO classification algorithm can overcome the defects of the clustering algorithms and improve the accuracy of fault classification.

4.2.2. Influence of randomness on the classification ability

In this section, we also did 50 experiments repeatedly under the same condition to get the statistical parameters of classification accuracies and verify the influence of randomness. Let the total number of particles $N_{\max} = 60$, the maximum iteration number $iter_{\max} = 1000$, and maturity period $S_1 = 200$.

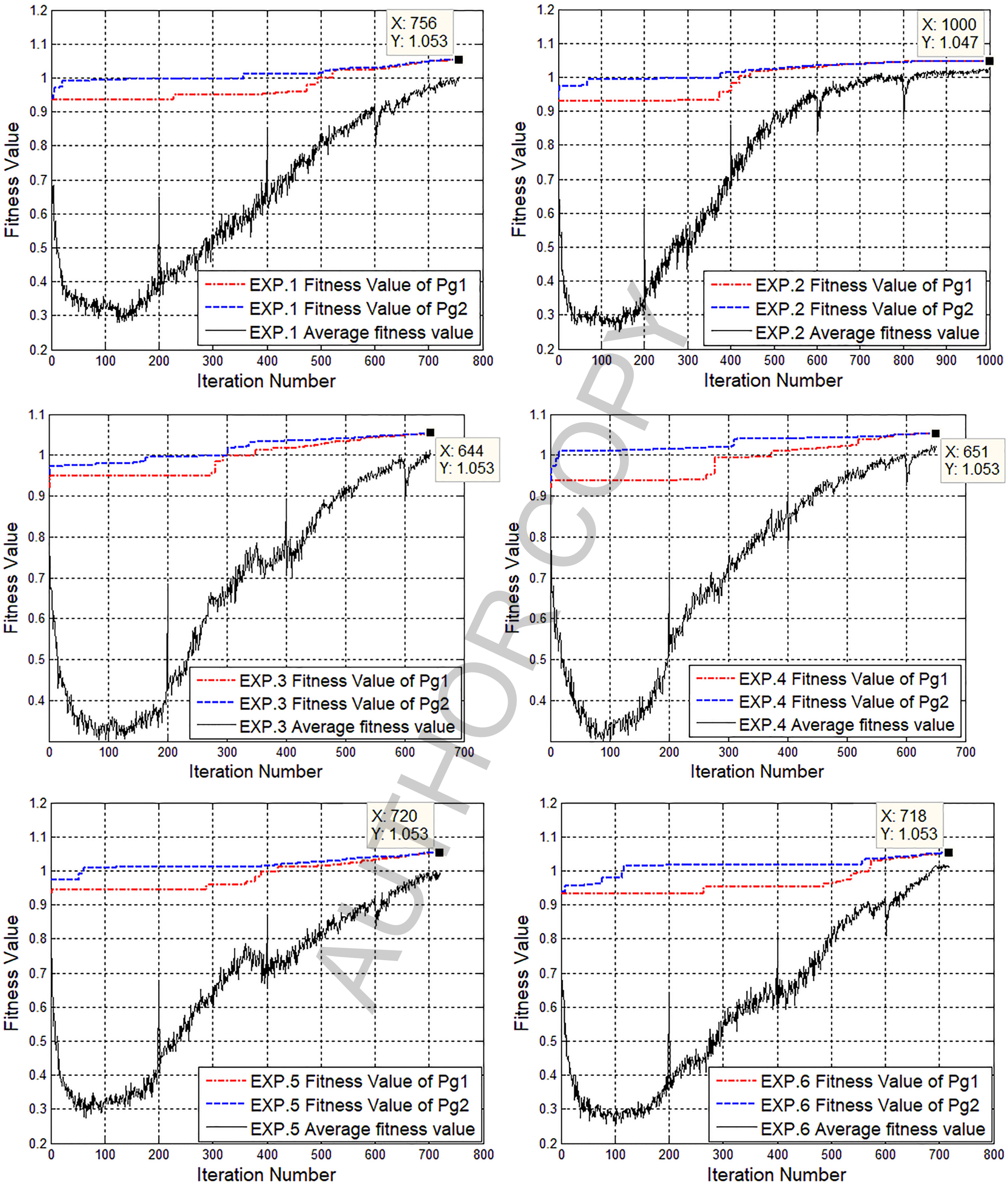

Fig. 7. Fitness change curves of first 6 experiments.

Table 5
Comparison of classification accuracy using different updating strategies

Models	Min	Mean	Max	STD
Model 1-1	0.8846	0.9838	1	0.0259
Model 1-2	0.8462	0.9368	1	0.0490
Model 2-1	0.8462	0.9363	1	0.0502
Model 2-2	0.8462	0.9346	1	0.0492
Model 3-1	0.7692	0.8885	1	0.0588
Model 3-2	0.7692	0.08869	1	0.0462

The particles from 1 to 38 are updated by the traditional updating strategy, and the particles from 54 to 60 are updated by the differentiation random positions updating strategy. The terminal conditions are $iter = iter_{\max}$ and the objective fitness value $V_f = 1.053$. Fifty different experiments were conducted. The results are shown in Fig. 6. It indicates that the classification results are very similar and have higher accuracy. There is small influence of randomness on this algorithm. Figure 7 is the change curve of $\text{fit}(\boldsymbol{P}_{g1})$, $\text{fit}(\boldsymbol{P}_{g2})$ and average fitness value of the first 6 experiments. At the early stage of iterations, the differentiation random positions updating can expand the local search ability, so that $\text{fit}(\boldsymbol{P}_{g1})$ is larger than $\text{fit}(\boldsymbol{P}_{g2})$, which increases the diversity of particles and improves the performance of the traditional positions updating strategy. At the later stage of iterations, when the particles converge to a certain area, a smaller value of inertia weight also is conducive to the local search. Therefore, $\text{fit}(\boldsymbol{P}_{g1})$ and $\text{fit}(\boldsymbol{P}_{g2})$ are similar. The updating strategies proposed in this paper have a good balance between the local search and the global search, which ensures that the S-PSO classification algorithm can overcome the influence of the randomness and farthest increase the probability of obtaining the global optimal solution.

4.2.3. Comparison of updating strategies

This section is to verify the classification ability of 6 models: (1) Model 1-1, S-PSO with hybrid positions updating strategy and intervention updating strategy, which has been calculated in Section 4.2.2; (2) Model 1-2, S-PSO with only hybrid positions updating strategy; (3) Model 2-1, PSO with traditional positions updating strategy and intervention updating strategy; (4) Model 2-2, PSO with traditional positions updating strategy; (5) Model 3-1, PSO with differentiation random positions updating strategy and intervention updating strategy; and (6) Model 3-2, PSO with only differentiation random positions updating strategy. The settings of these models are shown in Section 4.2.2 and these models are run under the same condition. Due to the influence of randomness on these models, we did 50 experiments for each model. The corresponding results are shown in Table 5. As shown in this table, the performance of Model 1-1 is optimal, the performance of Model 2-1 is suboptimal, and the performance of Model 3-2 is the worst. Table 5 also confirms the previous analysis about different updating strategies. The differentiation random positions updating strategy is not conducive to the global search and is very sensitive to the initial positions of the particles. Moreover, the intervention updating strategy can improve the performances of the classification models and increase the classification accuracy. These experiments further demonstrate the superiority of the S-PSO classification algorithm.

4.2.4. Comparison of different fitness functions

We also consider the influence of four fitness functions. Fitness function 1 only calculates the shorter distance of the intra-class, fitness function 2 only calculates the accuracy of training samples classification, fitness function 3 calculates both of the shorter distance of the intra-class and the longer distance of the inter-class, and fitness function 4 involves all of 3 factors. The fitness function 4 described in Section 4.2.2 are used. The setting of the algorithm is also shown in Section 4.2.2. The algorithms with

Table 6
Comparison of classification accuracy using different fitness functions

Fitness functions	Min	Mean	Max	STD
Function 1	0.8519	0.8519	0.8519	0
Function 2	0.7692	0.8681	0.9615	0.0503
Function 3	0.8740	0.8740	0.8740	0
Function 4	0.8846	0.9838	1	0.0259

Table 7
Comparison of classification accuracy using different algorithms

Algorithm	Min	Mean	Max	STD
S-PSO	0.8846	0.9838	1	0.0259
SVM	0.9231	0.9231	0.9231	0
BP	0.8846	0.9623	1	0.0352
LVQ	0.8846	0.8885	0.9231	0.0117
Bayes	0.8462	0.9631	1	0.0439
k-NN	0.8077 (k = 9, 10)	0.8808	1 (k = 1, 2)	0.0713

different fitness functions are also ran under the same condition. We did 50 experiments for each model. The results are listed in Table 6. It is obvious that the fitness function considering three factors is much more conducive to classify test samples than others.

4.2.5. Comparison with popular classification methods

The proposed S-PSO classification algorithm is also compared with other popular classification algorithms mentioned in Section 4.1. The S-PSO has been calculated in Section 4.2.2, and the settings for all algorithms are also as same as in Section 4.1. We did 50 experiments to compare their classification performance. Specially, the vales of k is changed from 1 to 10. The comparison results are shown in Table 7.

From this table, in terms of 80 samples, the classification accuracy of S-PSO algorithm is the highest, the classification accuracy of LVQ network is the lowest, and the SVM algorithm and LVQ network are more stable than other methods. The main reason is that the non-linear situation of these 80 samples in space distribution is more obvious. There are some overlaps among different classes, but the overlaps can be distinguished by the non-linear hyperplane or non-linear mapping. Hence, SVM algorithm and BP network have a good performance. The LVQ network only uses Euclidean distance to measure the difference of samples, which easily cause wrong classification to the overlap areas, so LVQ network has the worst performance. Although the k-NN algorithm can be calculated easily, it is sensitive to the selection of the parameter k, and the classification accuracy will be significantly changed with a minor change of parameter k, which degrade the performance of this algorithm.

5. Conclusions

Based on the above experimental results, the conclusions are drawn:

(1) In this paper, the prior class labels are used as supervised signals to construct the PSO classification algorithm, which overcomes the problems of commonly used clustering algorithms. S-PSO classification algorithm improves the classification accuracy and can satisfy the requirement of accurate fault diagnosis.

(2) Aiming at the classification problem and taking the distance from sample to class centers as the basis of classification, a fitness function based on shorter distance of intra-class, longer distance of inter-class and maximum classification accuracy of train samples is defined to make these three factors constraint the output optimal class centers. It enhances the generality and fault-tolerant ability of the classification algorithm in fault diagnosis, and increases the classification accuracy.
(3) The hybrid particle position updating strategy, which consists of differentiation random positions updating strategy and traditional positions updating strategy, is proposed to increase the diversity and enhance the global search ability. Meanwhile, the fixed iteration interval intervention updating strategy is also designed to make some particles adaptively jump out of current search area under certain conditions and search the entire solution space as much as possible. Two strategies are designed to ensure that the global optimal solution can be obtained with higher probability.

In one word, the S-PSO can correctly classify unknown fault samples from different machinery, so that fault causes, fault locations and fault levels can be determined accurately, thereby improving the efficiency of trouble-shoot, shortening the maintenance period, reducing the maintenance costs and ensuring machinery operation safety reliable.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under Grant ZYGX2014Z010, SKLMT-KFKT-201601, and the General Program of Civil Aviation Flight University of China under contract number J2015-39.

References

[1] P. Shakya, M.S. Kulkarni and A.K. Darpe, Bearing diagnosis based on Mahalanobis-Taguchi-Gram-Schmidt method, *Journal of Sound and Vibration* **337** (2015), 342–362.
[2] Y.F. Li, Z. Lv, W. Cai, S.P. Zhu and H.Z. Huang, Fatigue life analysis of turbine disks based on load spectra of aero-engines, *International Journal of Turbo & Jet Engines* **33** (2016), 27–33.
[3] Y.F. Li, H.Z. Huang, Y. Liu, N. Xiao and H. Li, A new fault tree analysis method: fuzzy dynamic fault tree analysis, *Eksploatacja i Niezawodnosc – Maintenance and Reliability* **14** (2012), 208–214.
[4] J. Mi, Y.F. Li, Y.J. Yang, W. Peng and H.Z. Huang, Reliability assessment of complex electromechanical systems under epistemic uncertainty, *Reliability Engineering & System Safety* **152** (2016), 1–15.
[5] H.Z. Huang, P. Cui, W. Peng, H. Gao and H.K. Wang, Fatigue lifetime assessment of aircraft engine disc via multi-source information fusion, *International Journal of Turbo & Jet Engines* **31** (2014), 167–174.
[6] D. Whitley, T. Starkweather and C. Bogart, Genetic algorithms and neural networks: optimizing connections and connectivity, *Parallel Computation* **14** (1990), 347–361.
[7] A.K. Jain, R.P.W. Duin and J.C. Mao, Statistical pattern recognition: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000), 4–37.
[8] B. Zheng and F. Gao, Research on the prediction of aeroengine wear based on the IPSO-SVR, *Lubrication Engineering* **39** (2014), 81–87.
[9] M.R. Tanweer and N. Sundararajan, Self regulating particle swarm optimization algorithm, *Information Science* **293** (2105), 182–202.
[10] Y. Song, Z. Chen and Z. Yuan, New chaotic PSO-based neural network predictive control for nonlinear process, *IEEE Transactions on Neural Networks* **18** (2007), 595–601.
[11] S. Suresh, P.B. Sujit and A.K. Rao, Particle swarm optimization approach for multi-objective composite box-beam design, *Composite Structures* **81** (2007), 598–605.
[12] A.K. Husseinzadeh, M.K. Husseinzadeh and S. Karimiyan, A particle swarm optimizer for grouping problems, *Information Sciences* **252** (2013), 81–95.
[13] L. Cagnina, M. Errecalde, D. Ingaramo and P. Rosso, An efficient particle swarm optimization approach to cluster short texts, *Information Sciences* **265** (2014), 36–49.

[14] Y.K. Lam, P.W.M. Tsang and C.S. Leung, PSO-based K-Means clustering with enhanced cluster matching for gene expression data, *Neural Computing & Applications* **22** (2013), 1349–1355.
[15] J. Avanija and K. Ramar, A hybrid approach using PSO and K-Means for semantic clustering of WEB documents, *Journal of Web Engineering* **12** (2013), 249–264.
[16] J. Zhang, Y.P. Wang and J.H. Feng, A hybrid clustering algorithm based on PSO with dynamic crossover, *Soft Computing* **18** (2014), 961–979.
[17] V. Fathi and G.A. Montazer, An improvement in RBF learning algorithm based on PSO for real time applications, *Neurocomputing* **111** (2013), 169–176.
[18] G.E. Guraksin, H. Hakli and H. Uguz, Support vector machines classification based on particle swarm optimization for bone age determination, *Applied Soft Computing* **24** (2014), 597–602.
[19] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* **2** (1998), 121–167.
[20] X.J. Wang, B.T. Zhang and D.Y. Xu, Compression Factor with grey model of particle swarm optimization and its application in smart grid, *Operatoins Research and Management Science* **21** (2012), 114–118.
[21] M.S. Leu and M.F. Yeh, Grey particle swarm optimization, *Applied Soft Computing* **12** (2012), 2985–2996.
[22] A. Ratnaweera, S.K. Halgamuge and H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation* **8** (2004), 240–255.
[23] F.V.D. Bergh and A.P. Engelbrecht, A cooperative approach to particle swarm optimization, *IEEE Transactions on Evolutionary Computation* **8** (2004), 225–239.
[24] S. Janson and M. Middendorf, A hierarchical particle swarm optimizer and its adaptive variant, *IEEE Transactions on Cybernetics* **35** (2006), 1272–1282.
[25] D. Parrott and X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, *IEEE Transactions on Evolutionary Computation* **10** (2006), 440–458.
[26] J. Brest, S. Greiner, B. Boskovic, M. Mernik and V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, *IEEE Transactions on Evolutionary Computation* **10** (2006), 646–657.
[27] B. Zheng, H.Z. Huang, H.W. Xu, D.B. Meng and X.L. Zhang, Multi-team competitive optimization algorithm and its application in bearing fault diagnosis, in: *2016 Annual Reliability and Maintainability Symposium*, 2016.
[28] R. Cheneg and Y.C. Jin, A social learning particle swarm optimization algorithm for scalable optimization, *Information Sciences* **291** (2015), 43–60.
[29] B. Zheng, Y.F. Li and H.Z. Huang, Aeroengine fault diagnosis method based on optimized supervised kohonen network, *Journal of Donghua University* **32** (2015), 1029–1033.
[30] S.J. Zinkle and Y. Matsukawa, Observation and analysis of defect cluster production and interactions with dislocations, *Journal of Nuclear Materials* **329** (2004), 88–96.
[31] C.H. Li and J.Z. Zhou, Semi-supervised weighted kernel clustering based on gravitational search for fault diagnosis, *ISA Transactions* **53** (2014), 1534–1543.
[32] B. Zheng and F. Gao, Fault diagnosis method based on S-PSO classification algorithm, *Acta Aeronautica et Astronautica Sinica* **36** (2015), 3640–3651.
[33] X.C. Liu, Entropy, distance measure and similarity measure of fuzzy sets and their relations, *Fuzzy Sets and Systems* **52** (1992), 305–318.
[34] S.X. Xiang, F.P. Nie and C.S. Zhang, Learning a mahalanobis distance metric for data clustering and classification, *Pattern Recognition* **41** (2008), 3600–3612.
[35] J. Ye, Cosine similarity measures for intuitionistic fuzzy sets and their applications, *Mathematical and Computer Modelling* **53** (2011), 91–97.
[36] A.M. Neto, Pearson's correlation coefficient: a more realistic threshold for applications on autonomous robotics, *Computer Technology and Application* **5** (2014), 69–72.
[37] A. Airteimoori and S. Kordrostami, A euclidean distance-based measure of efficiency in data envelopment analysis, *Optimization* **59** (2010), 985–996.
[38] S. Kiranyaz, T. Ince and A. Yildirim, Evolutionary artificial neural networks by multi-dimensional particle swarm optimization, *Neural Networks* **22** (2009), 1448–1462.
[39] S.J. Nanda and G. Panda, Automatic clustering algorithm based on multi-objective immunized PSO to classify actions of 3D human models, *Engineering Applications of Artificial Intelligence* **26** (2013), 1429–1411.
[40] F. van den Bergh, A analysis of particle swarm optimizers, Ph.D. Dissertation, Pretoria: University of Pretoria, 2002.
[41] B. Zheng, Investigation on aeroengine maintenance level decision on PSO-SVM, *Journal of Propulsion Technology* **34** (2013), 687–692.
[42] J.B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello and F. Fnaiech, Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration, *Applied Acoustics* **89** (2015), 16–27.
[43] B. Biswal, M. Biswal, S. Hasan and P.K. Dash, Nonstationary power signal time series data classification using LVQ

classifier, *Applied Soft Computing* **18** (2014), 158–166.
[44] M.M. Lange and N.A. Novikov, Bayes classifier based on tree-structured Gaussian mixtures, *Pattern Recognition and Image Analysis* **22** (2012), 136–143.
[45] J. Li, Research on the k-NN classification, *Natural Sciences Journal of Harbin Normal university* **29** (2013), 8–11.
[46] C.Z. Li and Y. Lei, Aircraft engine gas path fault diagnosis based on neural network, *Applied Mechanics and Materials* **148** (2012), 144–148.
[47] G. Chen and Y. Tang, Aero-engine interior damage recognition based on texture features of borescope image, *Chinese Journal of Scientific Instrument* **29** (2008), 1709–1713.
[48] Y. Tang, Research on Aero-engine interior damage evaluation and maintenance decisions based on internet, Ph.D. Dissertation, Nanjing: Nangjing University of Aeronautics and Astronautics, 2007.
[49] M. Eusuff, K. Lansey and F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, *Engineering Optimization* **38** (2006), 129–154.
[50] J.G. Jiang, L.Y. Zhang, Q. Su, L.J. Deng and M.N. Zhang, Shuffled frog leaping algorithm using dynamic searching strategy, *Journal of XiDian University* **41** (2014), 51–57.