A Novel Spectral Clustering Algorithm Based on Randomly State Changed Particle Swarm Optimization

Hao Chen, Dechun Guo, Shengzi Yang and Xiaochen Hou

School of Information and Electronics, Beijing Institute of Technology, Beijing, China
Email: Zjjnch123@163.com

Abstract. Spectral clustering algorithm is a method of clustering which allows one piece of data to belong to two or more clusters. In this paper, a novel spectral clustering algorithm based on randomly state changed particle swarm optimization is proposed. The initial population was classified by considering global and local optimal functions, the evolutionary state of each particle is considered through the comparison of cost functions, and the evolutionary state of the particles was subdivided. The Weight mode added the previously optimal local particles and global particles. According to the rule that newer particles have greater weights, particles speed was updated to reduce the possibility of falling into a local optimal state and to expand the search range of particles. The classification accuracy of the clustering algorithm was presented. Finally, by using the UCI datasets for comparison, it was found that the algorithm proposed in this paper increase the performances by 3% to 28% for different datasets, comparing with the known clustering algorithms such as particle swarm optimization algorithm, Fuzzy C-Means algorithm and conventional Spectral Clustering.

1. Introduction

Clustering is to classify the set of objects into multiple classes composed of similar objects. Clustering is one of the most popular methods in machine learning and image processing, and its applications range from statistics, biology and computer science to social science or psychology [1]. In data analysis, clustering algorithms are widely used because of the practicability and effectiveness, playing a very important role in the field. Spectral clustering is an innovative clustering technique used to partition graph matrices [2]. The basic algorithm of spectral clustering is divided into two steps. The first step is to calculate feature vectors by constructing a similarity matrix to reduce the dimension of the data and the complexity of the data. Then the second step, the K-means algorithm is used to cluster feature vectors into the required clusters. It makes an undirected weighted graph based on the data objects and obtains the best clustering results by considering the eigenvalues and eigenvectors associated with the graph.

However, the clustering accuracy of spectral clustering is also largely determined by cluster centroid. Therefore, some researchers adjust and optimize the K-means algorithm. For example, a spectral clustering based on global K-means (GKSC) was proposed by Xie [3], which use global K-means to reduce the initialization sensitivity. While, other researchers analyse the spectral clustering methods deeply. [4]. But the K-means algorithm is easy to give the local optimum. Therefore, some researchers use the Swarm Intelligence to achieve the global optimal solution in order to solute the problem of K-means [5].

Therefore, PSO has attracted our attention because of its simple concept and effectiveness. Many researchers are working on various methods to improve their search performance of PSO. One variation is the introduction of inertial weights W [6]. The results show that larger inertial weights tend to
contribute to overall exploration, while smaller inertial weights can enable local exploration to fine-tune the current search area [6]. Some researchers considered the density of the particle swarm [7]. Some researchers mentioned that adding time-delays in velocity updating process could help the performance of PSO [8-10]. Recently, other researchers proposed randomly occurring distributed delayed PSO algorithm to improve the accuracy of traditional clustering by using time-delay [11]. Although, the time-delays item can improve the effectiveness and robustness of PSO algorithm, it also be troubled by trapping in local optimal solution and adding time-delays means the complexity of calculating.

Therefore, in order to solve these problems, a novel randomly state changed particle swarm optimization (RSCPSO) algorithm is proposed to optimize spectral clustering. In this algorithm, the distributed delay terms are constrained by weight matrix and adding the elimination state will be helpful. The constrained time-delay items could help: (a) make full use of experience; (b) greatly reduce the possibility of falling into a local optimum; (c) expand the advantageous search range; (d) make the algorithm more robust. Therefore, the spectral clustering algorithm does not depend on initial clustering centres and can get global optimal solution, which can obtain better clustering results.

2. Methodology

2.1. Spectral Clustering Algorithm

Given a dataset X, the similarity matrix can be defined as $W = [W_{ij}]$, where $W_{ij}$ represents the correlation between the two points $X_i$ and $X_j$. The data is represented in an undirected graph $G = (V, E)$, where the vertex, $V$, represents all data points and $E$ represents the edges that connect the data. To achieve clustering effects, our goal is to divide the undirected graph $G$ into the subgraphs representing clusters, so that the correlations between the subgraphs are the lowest while the correlation within the graph is the highest. So that we can solve it by solving the eigenvalues and eigenvectors of the similarity matrix, which is usually defined as:

$$W(x, y) = e^{-\frac{|x-y|^2}{2\sigma^2}}$$

where $|x-y|^2$ represents the similar matrix between the datasets point $X$ and the datasets point $Y$. In this way, we can map each point to a low-dimensional space by finding the eigenvalues and eigenvectors of the matrix, reducing the complexity of the operation and making the clustering algorithm more effective. For spectral clustering, the most important part is the Tulaplacian matrix, and the general Tulaplacian matrix can be defined as follows:

In term of non-standardized Tulaplacian operator, the matrix is defined as

$$d_i = \sum_{j=1}^{n} W_{ij}$$

$$D = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix}$$

$$LM = (D, W)$$

Normalized Tulaplacian operator:

$$LM_{sm} = D^{-\frac{1}{2}}LMD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$$

$$L_{rw} = D^{-\frac{1}{2}}LM = I - D^{-\frac{1}{2}}W$$

where LM represents the Laplacian matrix, and $W$ is a weighted graph with a weight matrix, where the elements of $W$, $W_{ij} = W_{ji} \geq 0$. In this formula, $I$ is the identity matrix, $D$ is a diagonal matrix, $LM_{sm}$ is a symmetric matrix, and $L_{rw}$ is closely related to random walks [12].

After we find the similarity matrix, we calculate the feature vector $U_1, U_2, \ldots, U_k$ to simplify the data complexity, then we use the k-means algorithm to complete the clustering.
2.2. Particle Swarm Optimization

PSO is an optimization method widely used in the fields of function optimization and image processing. It can realize the information interaction within the particle swarm to achieve the intelligence in the problem-solving process. Particle swarm optimization is an overall optimization method, and each particle represents a possible solution to the current optimization problem. During each iteration, the particle’s acceleration and movement direction are determined by the particle’s current position and the group’s common experience. The optimization method represents that the particles will move towards the particle, to find the global best solution in moving when a particle finds the current best possible solution.

In PSO, an entire population consists of $G$ particles that search for movement in a $S$-dimensional space. A vector can represent the position of the $i$-th particle: $X_i(k) = (X_{i1}(k), X_{i2}(k), \ldots, X_{iS}(k))$. During the search process, the particle will adjust its position toward the global optimal direction of the group's common experience. The optimization method represents that the particles will move towards the particle, to find the global best solution in moving when a particle finds the current best possible solution.

It can realize the information interaction within the particle swarm to achieve the intelligence in the problem-solving process. Particle swarm optimization is an overall optimization method, and each particle represents a possible solution to the current optimization problem. During each iteration, the particle’s acceleration and movement direction are determined by the particle’s current position and the group’s common experience. The optimization method represents that the particles will move towards the particle, to find the global best solution in moving when a particle finds the current best possible solution.

The speed update model. More specifically, a particle represents a possible solution to the current optimization problem. During each iteration, the particle’s acceleration and movement direction are determined by the particle’s current position and the group’s common experience. The optimization method represents that the particles will move towards the particle, to find the global best solution in moving when a particle finds the current best possible solution.

We define $\omega$ as inertial weight that decreases linearly with iteration:

$$\omega = -(\omega_1 - \omega_2) \times \frac{iter}{maxiter} + \omega_1$$

where $\omega_1$ is the starting inertia proportional weight values and $\omega_2$ is the final inertia proportional weight values; $iter$ is the iteration numbers, and $maxiter$ is the maximum iteration numbers to control the end of algorithm. Usually we set the $\omega_1 = 0.8$ and the $\omega_2 = 0.5$ [12].

We define $c_1$ and $c_2$ as acceleration coefficients, which can be regarded cognitive and social parameters, respectively. The update formula of $c_1$ and $c_2$ are given in equations (8) and (9), respectively, where $c_{1e}$ and $c_{2e}$ are the end value of acceleration coefficients. $c_{1i}$ and $c_{2i}$ are the initial value of acceleration coefficients.

$$c_1 = -(c_{1i} - c_{1e}) \times \frac{iter}{maxiter} + c_{1i}$$

$$c_2 = -(c_{2i} - c_{2e}) \times \frac{iter}{maxiter} + c_{2i}$$

Therefore, we use the following formula to calculate the next speed and position of the particle in the particle swarm:

$$V_i(t + 1) = \omega(t)v_i(t) + c_1 rand() (P_i - x_i(t)) + c_2 rand() (G - x_i(t))$$

$$x_i(t + 1) = v_i(t + 1) + x_i(t)$$

where $\omega(t)$ is the proportional weight, $t$ means the iteration numbers. $rand()$ means random number between 0 and 1.

2.3. SCRSCPSO

We proposed a new PSO algorithm to improve the clustering effectiveness—a spectral clustering algorithm based on randomly state changed particle swarm optimization (SCRSCPSO). The main novelty of this algorithm is that it can introduce the time delay according to the weight distribution into the speed update model. More specifically, a particle’s cost function is used to determine the evolutionary state of the particle, and then select different coefficients of the historically optimal particles and global optimal particles based on different evolutionary states.
In the proposed algorithm, the update of the speed and the position in the \( k \)-th iteration can be formulated as
\[
v_i(t + 1) = \omega(t)v_i(t) + \text{rand}(\omega) c_1 (P_i(t) - x_i(t)) + \text{rand}(\omega) c_2 (P_g(t) - x_i(t)) + \text{rand}(\omega) c_3 \sum_{\tau=1}^{N} (\frac{1}{2})^\tau (P_i(k - \tau) - x_i(k)) + \text{rand}(\omega) c_4 \sum_{\tau=1}^{N} (\frac{1}{2})^\tau (P_g(k - \tau) - x_i(k))
\]
(12)

\[
x_i(t + 1) = v_i(t + 1) + x_i(t)
\]
(13)

In equation (12), \( \omega \) is inertia proportional weight which is defined below; we can calculate the \( c_1 \) and \( c_2 \) according to equation (8) and equation (9). \( c_3 \) and \( c_4 \) are acceleration coefficients of past Gbest and Pbest experience delay term and we set \( c_1 = c_3, c_2 = c_4; N \) represents the total number of past experience; \((1/2)^t\) is an \( N \) dimension vector as weight item. Each element of this vector is determined by the distributed delay. Since the weight item makes the old experience items play an auxiliary role and makes the latest experience items play a major role. \( t_i(\mu) \) and \( t_g(\mu) \) which \( t_i(\mu) \) represents the intensity factors of the local optimum of delay term distributed and \( t_g(\mu) \) means the global optimum of delay term distributed according to the evolution state \( \mu \). In term of the principle of greedy search, \( \text{rand}(\omega) \) is random number from 0 to 1 that ensures the search step size in the search process will not be too large, so as not to miss the best solution. This can reduce search time and improve efficiency.

The selection of inertia weight and acceleration coefficient is very important for the realization of RSCPSO algorithm. We adjust the inertial weights to improve local and global search capabilities.

In the proposed RSCPSO algorithm, the velocity and position equations are updated according to the evolution state, depending on the evolution factors, as described in Refs. [1, 2]. The search feature of the PSO algorithm are the search state, development status, inquiry status and exit status, and elimination status. To reveal the five evolutionary states, we use \( \mu(k) = A, \mu(k) = B, \mu(k) = C, \mu(k) = D, \) and \( \mu(k) = E \) to represent five states, respectively.

We calculate the evolution factor based on the distance between particles. The average fitness of the \( i \)-th particle expressed by \( D_i \) as follows:
\[
D_i = \sum_{j=1,j \neq i}^{S} \sum_{k=1}^{G} d|x_{ik} - x_{jk}|
\]
(14)

In this formula, \( S \) is the cluster size and \( G \) is the dimension size. The evolutionary factors represented by \( H_f \) are as follows:
\[
H_f = \frac{D_i - D_{\text{min}}}{D_{\text{max}} - D_{\text{min}}}
\]
(15)

where \( D_i \) represents the average fitness of \( i \)-th particle; \( D_{\text{min}} \) and \( D_{\text{max}} \) represent the minimum and maximum values of \( D_i \), respectively. In this paper, \( \mu(k) \) represents an evolution state by using the equal division strategy as follows:

\[
\mu(k) = \begin{cases} 
A, & 0.00 \leq H_f < 0.25 \\
B, & 0.25 \leq H_f < 0.5 \\
C, & 0.50 \leq H_f < 0.75 \\
D, & 0.75 \leq H_f < 1.00 \\
E, & \text{else} 
\end{cases}
\]
(16)

In the collection state represented by \( \mu(k) = A \), the particles try to fly into the global optimal region as soon as possible. Therefore, the speed update model in the traditional PSO algorithm is adopted, and by setting the intensity factor to zero, which means \( t_i(\mu) = 0 \) and \( t_g(\mu) = 0 \). The distributed delay term is ignored.

In the development state represented by \( \mu(k) = B \), it is assumed that the particles can develop the area around the personal best particles. In order to avoid premature convergence, a randomly occurring
distributed delay is added to the speed update model, and a certain number of historical personal best particles are randomly selected for a more thorough search. In this case, the intensity factor is set to

\[ t_i(\mu) = 0.05 \times (1 + e^{\text{iter} \cdot \mu_{\text{maxiter}}})^{-1}, \text{ and } t_g(\mu) = 0 \]

In the detection state represented by \( \mu(k) = C \), particles are encouraged to thoroughly explore the entire search space. Therefore, we need to add the random distribution time delay to the speed update model, and use the intensity factor to add the historical global optimal particles to our speed update formula according to formula: \( t_i(\mu) = 0, \text{ and } t_g(\mu) = 0.05 \times (1 + e^{\text{iter} \cdot \mu_{\text{maxiter}}})^{-1} \). The constraints of particles and distribution delays allow particles to explore and decompose more quickly with better exploration strategies to find the overall optimal point.

In the escape state represented by \( \mu(k) = D \), the particles need to get rid of the local optimum and move in the direction of global optimum. Therefore, we continue to add distributed delays in the speed update model to get rid of local optimality through past historical experience. Therefore, in this model, a certain number of historical personal best particles and global best particles are randomly selected with intensity factors \( t_i(\mu) = 0.05 \times (1 + e^{\text{iter} \cdot \mu_{\text{maxiter}}})^{-1}, \text{ and } t_g(\mu) = 0.05 \times (1 + e^{\text{iter} \cdot \mu_{\text{maxiter}}})^{-1} \). Through distributed delays, historical personal best particles and global best particles have common constraints to help the particles out of this state faster.

In the elimination state represented by \( \mu(k) = E \), based on the greedy selection strategy, we eliminate the particles with the largest deviation and re-select a new particle from the data set to avoid large deviation particles from affecting the convergence of the entire particle swarm. Therefore, the particle swarm can get the global best solution faster.

The proposed SCRSCPSO algorithm is explained in the following systematic process:

**Algorithm 1: SCRSCPSO**

**Input:** dataset \( X = \{X_1, X_2, \cdots, X_n\} \), cluster number \( k \).

**Output:** the cluster \( C = \{C_1, C_2, \cdots, C_k\} \).

**Step1:** Calculate similarity matrix \( W \) of dataset \( X \) using the formulas from equation (1).

**Step2:** Obtain the degree matrix \( D \) by using formula equation (2) to equation (3).

**Step3:** Normalize similarity matrix by using equation (4) to equation (5) and calculate the eigenvectors of the Laplacian matrix.

**Step4:** Find the largest eigenvector of the Normalize the Laplacian matrix and then generate a \( k \) -dimensional matrix \( V = \{V_1, V_2, \cdots, V_k\} \), where \( k \) is the number of clusters.

**Step5:** Initialize the particle swarm parameters, randomly selecting \( k \) cluster centers for classification and calculating the personal best \( P_i \) and global best \( P_g \).

**For iter < maxiter:**

**Step6:** Calculate the \( D_i \) of particles separately by using equation (14).

if \( (D_i < P_i) \)

\[ P_i = D_i \]

if \( (P_i < P_g) \)

\[ P_g = P_i \]

**Step7:** Calculate the \( H_f \) using equation (15) and determining the next state of particle according to equation (16).

**Step8:** Update inertia weight \( \omega \) and acceleration coefficients \( c_1, c_2, c_3, c_4, t_g(\mu) \) and \( t_i(\mu) \) according to the number of iterations and update the historical local optimal and global local optimal weight factors according to equation (7) to equation (9) and table 1.

**Step9:** Update particle velocity and position using equation (12) to equation (13).

\[ \text{iter} = \text{iter} + 1 \]

**End for**

**Step10:** Select the global best particle and clustering datasets by using its clustering centroids.
Step11: output the clustering results.

| State       | Mode | $t_f(\mu)$ | $t_g(\mu)$ |
|-------------|------|------------|------------|
| Collection  | $\mu(k) = A$ | 0          | 0          |
| Development | $\mu(k) = B$ | $0.05 \times (1 + e^{\frac{\text{iter}}{\text{maxiter}}})^{-1}$ | 0          |
| Detection   | $\mu(k) = C$ | 0          | $0.05 \times (1 + e^{\frac{\text{iter}}{\text{maxiter}}})^{-1}$ |
| Escape      | $\mu(k) = D$ | $0.05 \times (1 + e^{\frac{\text{iter}}{\text{maxiter}}})^{-1}$ | $0.05 \times (1 + e^{\frac{\text{iter}}{\text{maxiter}}})^{-1}$ |
| Elimination | $\mu(k) = E$ | 0          | 0          |

3. Simulation

All the experiments were carried out on a 2.4Ghz Intel CPU with 16GB of memory, running on windows 10 and MATLAB2016. We run the algorithm to cluster the data set and used the results to detect the performance of the algorithm and the quality of the clustering results.

We used this algorithm in the UCI datasets, through a large number of data tests to ensure the validity and rationality of the experiment. We also compared this algorithm with ordinary PSO algorithm, ordinary spectral clustering algorithm, and FCM algorithm to verify whether the clustering algorithm is optimized. The metrics we choose were accuracy, cross-correlation and mutual information metric, so we could check whether the clustering effectiveness has reached the optimum and whether our algorithm has been optimized for the global optimal solution.

3.1. Datasets

In this experimentation, we used many data sets from different fields to ensure that there is no correlation between the data sets and to ensure the independence of each experiment. The datasets we used as follows:

(1) Ecoli datasets; (2) Dermatology datasets; (3) Segments datasets; (4) Wine datasets; (5) Seed datasets.

In addition, the clustering performance was assessed by the performance measures which was named Accuracy and Normalized Mutual Information (NMI). We could compare our algorithm with other known algorithm using the measures and the performance measures was defined as follow:

3.2. Accuracy

We define accuracy as the ratio of accurate matching pair number to total matching pair number. The accurate matching pair number can be defined as $TP + TN$ and the total matching pair number can be defined as $TP + TN + FP + FN$. $TP$ means clustering two similar data points in the same cluster. $TN$ means clustering two different cluster data points in the different cluster. $FP$ means clustering two different cluster data points in the same cluster. $FN$ means clustering two similar cluster data points in the different cluster. Therefore, the accuracy can be calculated by the following formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

3.3. Normalized Mutual Information (NMI)

Standardized mutual information, which is used to reflect the quality of clustering, means the information overlap degree of two clusters. Therefore, we normalized the mutual information to compare various numbers of clusters to verify the performances of this algorithm. Let $K$ be the set of clusters, $C$ be the class label, $H(X)$ is entropy and $I(C:K)$ is the mutual information between $C$ and $K$ [12]. Then mutual information is calculated by using

$$I(C; K) = H(C) - H(C|K) \quad (18)$$
3.4. Datasets Experiments
In this section, we used many datasets on UCI to do experiments. Through many experiments to verify the effectiveness of our algorithm.

These datasets were commonly used, which were generally accepted by clustering algorithms, so the reliability of the datasets could be guaranteed. Then we compared our algorithm with conventional spectral clustering and conventional particle swarm clustering algorithms. Tables 2 and 3 summarized the accuracy and NMI performance of these algorithms on different data sets. By comparison, we could find that our algorithm is superior to the conventional clustering algorithm, reflecting the superiority of our algorithm.

Table 2 compared the accuracy of the clustering results of several clustering algorithms for different datasets. From this, we could find that our SCRSCPSO algorithm is superior to other algorithms, such as Spectral Clustering algorithm, Fuzzy C-Means algorithm, and PSO algorithm, since, SCRSCPSO algorithm increased its performances by 3% to 5% for Ecoli dataset comparing with other algorithms and increased its performances by 3% to 9% for Dermatology dataset. For seed dataset, SCRSCPSO algorithm provided 1% optimization. For Segments datasets, SCRSCPSO algorithm increased the accuracy by 2% to 8% and increases the accuracy by 2% to 28% for wine datasets.

Table 3 compared the NMI index of the clustering results of several clustering algorithms for the datasets. From the table, we could find that our SCRSCPSO algorithm increased the performances by 2% to 6% for Ecoli dataset. For Dermatology dataset, our SCRSCPSO got 2% to 17% growth comparing with other algorithms. SCRPSO algorithm also increased its performances by 4% to 15% for Segments datasets. For wine dataset, it increased its performances by 1% to 40%. For Seed dataset, SCRSCPSO algorithm provided 1% to 2% optimization.

Table 2. The accuracy of clustering datasets.

|           | Ecoli | Dermatology | Segments | Wine | Seed |
|-----------|-------|-------------|----------|------|------|
| SCRSCPSO  | 0.800 | 0.977       | 0.865    | 0.924| 0.879|
| Spectral Clustering | 0.773 | 0.949 | 0.845 | 0.908 | 0.870 |
| PSO       | 0.772 | 0.920       | 0.860    | 0.891| 0.867|
| FCM       | 0.761 | 0.893       | 0.786    | 0.641| 0.869|

Table 3. The NMI of clustering datasets.

|           | Ecoli | Dermatology | Segments | Wine | Seed |
|-----------|-------|-------------|----------|------|------|
| SCRSCPSO  | 0.666 | 0.939       | 0.653    | 0.818| 0.691|
| Spectral Clustering | 0.646 | 0.919 | 0.612 | 0.805 | 0.682 |
| PSO       | 0.613 | 0.825       | 0.609    | 0.804| 0.670|
| FCM       | 0.604 | 0.764       | 0.505    | 0.404| 0.673|

4. Conclusion
Nowadays clustering is a very popular issue in machine learning field [13]. Therefore, we summary other algorithms to find their advantages and disadvantages. To solve the problems and improve the effectiveness of clustering algorithm, a novel algorithm SCRSCPSO is proposed.

According to the simulation, we could find that the performance of our SCRSCPSO algorithm was better than the other current algorithm. We could get the global optimal solution by using this algorithm proposed in this article. Since for a certain particle, the weight coefficient of the time-delay items and
acceleration decreased with time. In this way, compared with the traditional delayed PSO algorithm and conventional Spectral Clustering, the time delay with the distribution of weight coefficients which we introduced allows us to: (1) make better use of the cumulative history of particle evolution, (2) make the particle swarm more stable and avoid falling into a local optimum, (3) ensure the diversity of particle search directions, (4) make the particle swarm search ability stronger, find the global optimum faster, and complete clustering. Therefore, compare with the conventional spectral clustering. A better optimal algorithm is proposed to optimize the spectral clustering and we can get an accurate clustering result by using this algorithm.

The SCRSCPSO can take advantage of time-delay item and make the search process of the particle more robust and efficient, which can help us get the optimized clustering results.

In the future, the time-delay item can be defined accurately. We can propose a detailed strategy to define the state of the particles. Therefore, the moving of particle can be more intelligent and we will get the global optimum faster. This algorithm could be used not only in machine learning field but for modeling fields such as architecture. I believe this algorithm has a lot of potential for development.

References
[1] Mohammed A J, Yusof Y and Husni H 2015 Document clustering based on firefly algorithm Journal of Computer Science 11 (3) 453-465.
[2] Kumar A and Daumé H 2011 A co-training approach for multi-view spectral clustering Proceedings of the 28th International Conference on Machine Learning (ICML-11) pp 393-400.
[3] Xie H, Zhang P W and Luo S 2010 Spectral clustering based on global k-means Journal of Computer Applications 30 (7) 1936-1937.
[4] Oliveira S and Seok S C 2005 A multi-level approach for document clustering International conference on computational science pp 204-211.
[5] Karaboga D and Ozturk C 2011 A novel clustering approach: Artificial Bee Colony (ABC) algorithm Applied soft computing 11 (1) 652-657.
[6] Shi J and Malik J 2000 Normalized cuts and image segmentation Departmental Papers IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) 888-905.
[7] Song B, Wang Z and Zou L 2017 On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm Cognitive Computation 9 (1) 5-17.
[8] Tang Y, Wang Z and Fang J 2011 Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm Expert Systems with Applications 38 (3) 2523-2535.
[9] Zeng N, Wang Z, Zhang H and Alsultan F E 2016 A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay Cognitive Computation 8 (2) 143-152.
[10] Ratnaweera A, Halgamuge S K and Watson H C 2004 Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients IEEE Transactions on Evolutionary Computation 8 (3) 240-255.
[11] Liu W, Wang Z, Liu X, Zeng N and Bell D 2019 A novel particle swarm optimization approach for patient clustering from emergency departments IEEE Transactions on Evolutionary Computation 23 (4) 632-644.
[12] Janani R and Vijayarani S 2019 Text document clustering using spectral clustering algorithm with particle swarm optimization Expert Systems with Applications 134 192-200.
[13] Selim S Z and Alsultan K 1991 A simulated annealing algorithm for the clustering problem Pattern Recognition 24 (10) 1003-1008.