Advances in Life Science and its Applications (ALSA)
Vol. 1, No. 2, 2012

www.worldsciencepublisher.org

Cluster Analysis by Variance Ratio Criterion and PSOSQP Algorithm

Yudong Zhang, Lenan Wu

(School of Information Science and Engineering, Southeast University, Nanjing China)

zhangyudongnuaa@gmail.com, wuln@seu.edu.cn

Abstract: In order to solve the cluster analysis problem more efficiently, we presented a new approach based on Particle Swarm Optimization Sequence Quadratic Programming (PSOSQP). First, we created the optimization model using the variance ratio criterion (VRC) as fitness function. Second, PSOSQP was introduced to find the maximal point of the VRC. The experimental dataset contained 400 data of 4 groups with three different levels of overlapping degrees: non-overlapping, partial overlapping, and severely overlapping. We compared the PSOSQP with genetic algorithm (GA) and combinatorial particle swarm optimization (CPSO). Each algorithm was run 20 times. The results showed that PSOSQP could found the largest VRC values among all three algorithms, and meanwhile it cost the least time. It can conclude that PSOSQP is effective and rapid for the cluster analysis problem.

Keywords: Cluster Analysis; Variance Ratio Criterion; Genetic Algorithm; Particle Swarm Optimization; Sequence Quadratic Programming.

1 Introduction

Cluster analysis is the assignment of a set of observations into subsets without any priori knowledge so that observations in the same cluster are similar to each other than to those in other clusters [1, 2]. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields [3], including machine learning [4], data mining [5], pattern recognition [6], feature reduction [7], decision tree [8], image analysis [9] and bioinformatics [10]. Cluster analysis can be achieved by various algorithms that differ significantly. Those methods can be basically classified into four categories:

I. Hierarchical Methods. They find successive clusters using previously established clusters. They can be further divided into the agglomerative methods and the divisive methods [11]. Agglomerative algorithms start with one-point clusters and recursively merges two or more most appropriate clusters [12]. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters [13].
II. Partition Methods. They generate a single partition of data with a specified or estimated number of non overlapping clusters, in an attempt to recover natural groups present in the data [14].
III. Density-based Methods. They are devised to discover arbitrary-shaped clusters. In this approach, a cluster is regarded as a region in which the density of data objects exceeds a threshold. DBSCAN [15] is the typical algorithm of this kind.
IV. Subspace Methods. They look for clusters that can only be seen in a particular projection (subspace, manifold) of the data. These methods thus can ignore irrelevant attributes [16].

In this study, we focus our attention on Partition Clustering methods. The *K*-means clustering [17] and the fuzzy *c*-means clustering (FCM) [18] are two typical algorithms of this type. They are iterative algorithms and the solution obtained depends on the selection of the initial partition and may converge to a local minimum of criterion function value if the initial partition is not properly chosen [19]. Branch and bound algorithm was proposed to find the global optimum clustering. However, it takes too much computation time [20].

In the last decade, evolutionary algorithms were proposed to clustering problem since they are not sensitive to initial values and able to jump out of local minimal point. For example,

Elcio Sabato de Abreu e Silva et al. [21] proposed the application of a genetic algorithm (GA) for determining global minima to be used as seeds for a higher level ab initio method analysis such as density function theory (DFT). Water clusters were used as a test case and for the initial guesses four empirical potentials (TIP3P, TIP4P, TIP5P and ST2) were considered for the GA calculations. Two types of analysis were performed namely rigid (DFT_RM) and non rigid (DFT_NRM) molecules for the corresponding structures and energies. For the DFT analysis, the PBE exchange correlation functional and the large basis set A-PVTZ had been used. All structures and their respective energies calculated

through the GA method, DFT_RM and DFT_NRM are compared and discussed. The proposed methodology showed to be very efficient in order to have quasi accurate global minima on the level of ab initio calculations and the data are discussed in the light of previously published results with particular attention to $(H_2O)n$ clusters. Lin et al. [22] pointed out that *k*-Anonymity has been widely adopted as a model for protecting public released microdata from individual identification. Their work proposed a novel genetic algorithm-based clustering approach for *k*-anonymization. Their proposed approach adopted various heuristics to select genes for crossover operations. Experimental results showed that their approach can further reduce the information loss caused by traditional clustering-based *k*-anonymization techniques. Chang et al. [23] proposed a new clustering algorithm based on genetic algorithm (GA) with gene rearrangement (GAGR), which in application may effectively remove the degeneracy for the purpose of a more efficient search. They used a new crossover operator that exploited a measure of similarity between chromosomes in a population. They also employed adaptive probabilities of crossover and mutation to prevent the convergence of the GAGR to a local optimum. Using the real-world data sets, they compared the performance of GAGR clustering algorithm with *K*-means algorithm and other GA methods. Their experiment results demonstrated that the GAGR clustering algorithm had high performance, effectiveness and flexibility. Agard et al. [24] pointed out defining an efficient bill of materials for a family of complex products was a real challenge for companies, largely because of the diversity they offered to consumers. They solution is to define a set of components (called modules), each of which contained a set of primary functions. An individual product was then built by combining selected modules. The industrial problem leads, in turn, to the complex optimization problem. They solved the problem via a simulated annealing method based on a clustering approach. Jarboui et al. [14] presented a new clustering approach based on the combinatorial particle swarm optimization (CPSO) algorithm. Each particle was represented as a string of length *n* (where *n* is the number of data points), and the *i*th element of the string denoted the group number assigned to object *i*. An integer vector corresponded to a candidate solution to the clustering problem. A swarm of particles were initiated and fly through the solution space for targeting the optimal solution. To verify the efficiency of the proposed CPSO algorithm, comparisons with a genetic algorithm were performed. Computational results showed that their proposed CPSO algorithm was very competitive and outperforms the genetic algorithm. Niknam et al. [25] considered the *k*-means algorithm highly depended on the initial state and converged to local optimum solution. Therefore, they presented a new hybrid evolutionary algorithm to solve nonlinear partitional clustering problem. Their proposed hybrid evolutionary algorithm was the combination of FAPSO (fuzzy adaptive particle swarm optimization), ACO (ant colony optimization) and *k*-means algorithms, called FAPSO-ACO-K, which can find better cluster partition. The performance of their proposed algorithm was evaluated through several benchmark data sets. Their simulation results showed that the performance of the proposed algorithm was better than other algorithms such as PSO, ACO, simulated annealing (SA), combination of PSO and SA (PSO-SA), combination of ACO and SA (ACO-SA), combination of PSO and ACO (PSO-ACO), genetic algorithm (GA), Tabu search (TS), honey bee mating optimization (HBMO) and *k*-means for partitional clustering problem. Roy Gelbard, et al. [26] considered cross-cultural research as a case in point and applied Multi-Algorithm Voting (MAV) methodology to cluster analysis. Their study was designed to provide more systematic supportive decision tools for researchers and managers alike when attempting to cluster analyzing phenomena. To assess the merits of the methodology of MAV for cluster analysis, they analytically examined cross-cultural data from Merritt study as well as data scored and ranked by Hofstede. Their study contributed to the literature in several ways. From a methodological point of view, they showed how researchers can avoid arbitrary decisions indetermining the number of clusters. They provided the researcher with more compelling and robust methodologies not only for analyzing the results of cluster analysis, but also for more better-grounded decision-making through which theoretical insights and implications can be drawn.

However, those aforementioned algorithms suffer from following shortcomings. They converged too slow, or even converged to local minima points, which lead to a wrong solution. Therefore, in this paper we introduced in the Particle Swarm Optimization Sequence Quadratic Programming (PSOSQP) algorithm [27] for optimization. Zhang et al. proposed the PSOSQP and proved it outperforms GA, PSO, and ABC [27].

The structure of the rest of this paper was organized as follows. Next section 2 defined the partitional problem, and gave the encoding strategy and clustering criterion. Section 3 introduced the Particle Swarm Optimization Sequence Quadratic Programming (PSOSQP) Algorithm. Experiments in section 4 contained three types of artificial data with different overlapping degree. Final section 1 was devoted to conclusions and future works.

2 Partitional Clustering Problem

The problem of partitional clustering can be depicted as follows. Suppose there are *n* samples $\boldsymbol{O} = \{o_1, o_2, \ldots, o_n\}$ in a *d*-dimensional metric space. Those samples are to be clustered into *k* groups so that the objects in a cluster are more similar to each other than to objects in different groups [28]. Each $o_i \in R^d$ represents a feature vector consisting of *d* real valued measures describing the feature of the objects. Suppose the clusters are denoted as $\boldsymbol{C} = \{c_1, c_2, ..., c_k\}$, then they should obey three following statements

$$\begin{aligned} & c_i \neq \phi \quad \text{for } i = 1, 2, ..., k \\ & c_i \bigcap c_j = \phi \quad \text{for } i \neq j \\ & \bigcup c_i = \{1, 2, ..., n\} \end{aligned} \tag{1}$$

Our task is to find the optimal partition $\boldsymbol{C}^*$ that has the best adequacy in terms to all other feasible solution. Two related issues need to be solved for translating the

clustering problem into an optimization problem. One is the encoding strategy, and the other is the criterion function.

2.1 Encoding Strategy

The search space is determined of *n*-dimension due to *n*-objects. Each dimension represents an object and the *i*th individual $X_i = \{x_{i1}, x_{i2}, ..., x_{in}\}$ corresponds to the affection of *n* objects, such that $x_{ij} \in \{1, 2, ..., k\}$, where *k* denotes the number of classes, *j* denotes the *j*th object, and *i* denotes the *i*th individual. Suppose *n*=9, *k*=3, the first cluster contains the 1st, 4th, and 7th object, the second cluster contains the 2nd, 5th, and 8th object, and the third cluster contains the 3rd, 6th, and 9th object. The encoding of this cluster solution is illustrated in Fig. 1.

Cluster	Object
I	1,4,7
II	2,5,8
III	3,6,9

	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	x_{i6}	x_{i7}	x_{i8}	x_{i9}
X_i	1	2	3	1	2	3	1	2	3

Fig. 1 An example of encoding representation

2.2 Criterion Function

There are several criteria that had been proposed to measure the adequacy or similarity in which a given data set can be clustered. The most common used partitional clustering strategy is the Variance Ratio Criterion (VRC) [29]. Its definition is as formulated

$$VRC = \frac{B}{W} \times \frac{n-k}{k-1} \tag{2}$$

Here ***W*** and ***B*** denote the within-cluster and between-cluster variations, respectively. They are defined as:

$$W = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left(o_j^i - \bar{o}_j\right)^T \left(o_j^i - \bar{o}_j\right) \tag{3}$$

$$B = \sum_{j=1}^{k} n_j \left(\bar{o}_j - \bar{o}\right)^T \left(\bar{o}_j - \bar{o}\right) \tag{4}$$

Where n_j denotes the cardinal of the cluster c_j, o_j^i denotes the *i*th object assigned to the cluster c_j, $\bar{o}_j$ denotes the *n*-dimensional vector of sample means within *j*th cluster (cluster centroid), and $\bar{o}$ denotes the *n*-dimensional vector of overall sample means (data centroid) [30]. (*n*-*k*) is the degree of freedom of the within-cluster variations, and (*k*-1) is the degree of freedom of the between-cluster variations [31].

As a consequence, compact and separated clusters are expected to have small values of *W* and large values of *B*. Hence, the better the data partition, the greater the value of VRC. The normalization term (*n*-*k*)/(*k*-1) prevents the ratio to increase monotonically with the number of clusters, thus making VRC as an optimization (maximization) criterion.

3 PSOSQP Algorithm

We introduced in the PSOSQP algorithm [27] in this section. The algorithm integrates PSO technique with SQP. PSO is the main algorithm, and meanwhile SQP is used to finely tune every step of the solution by PSO. PSO has a more global searching ability at the beginning of the run, but a bit ineffective for a local search near the end of the run. To overcome this drawback, we integrate PSO with a gradient search algorithm called SQP. In the beginning of the run, PSO has more possibilities to explore a large space and therefore the particles are easier to move and jump out of local minima. The best value of all the particles will be taken as the initial starting point for the SQP and will be finely tuned. The possibility of exploring a global minimum with more local optima is increased. The search will continue until a termination criterion is satisfied.

3.1 Particle Swarm Optimization

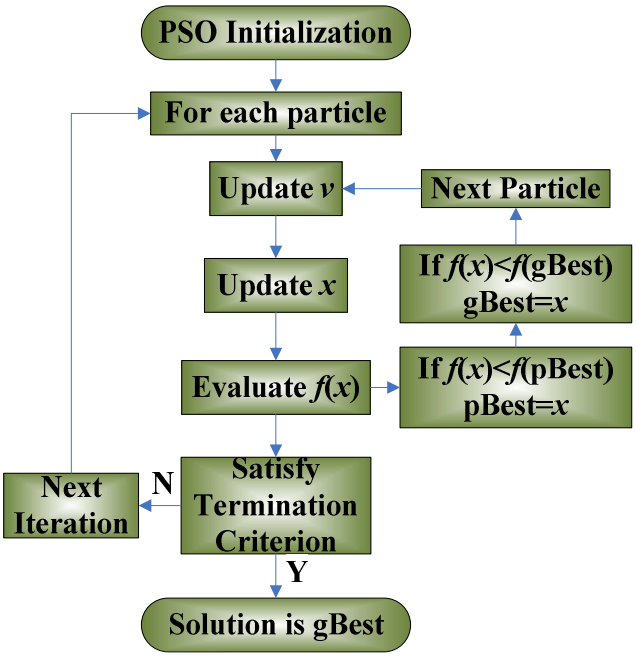

Fig. 2 Flow chart of the PSO algorithm

PSO is a population based stochastic optimization technique, which simulates the social behavior of a swarm of bird, flocking bees, and fish schooling. By randomly initializing the algorithm with candidate solutions, the PSO successfully leads to a global optimum. This is achieved by an iterative procedure based on the processes of movement and intelligence in an evolutionary system. Fig. 2 shows the flow chart of a PSO algorithm.

In PSO, each potential solution is represented as a particle. Two properties (position x and velocity v) are associated with each particle. Suppose x and v of the ith particle are given as

$$x = (x_{i1}, x_{i2}, \cdots, x_{iN}) \tag{5}$$

$$v = (v_{i1}, v_{i2}, \cdots, v_{iN}) \tag{6}$$

where N stands for the dimensions of the problem. In each iteration, a fitness function is evaluated for all the particles in the swarm. The velocity of each particle is updated by keeping track of the two best positions. One is the best position a particle has traversed so far and called "*pBest*". The other is the best position that any neighbor of a particle has traversed so far. It is a neighborhood best called "*nBest*". When a particle takes the whole population as its neighborhood, the neighborhood best becomes the global best and is accordingly called "*gBest*". Hence, a particle's velocity and position are updated as follows

$$v = \omega \cdot v + c_1 r_1 (pBest - x) + c_2 r_2 (nBest - x) \tag{7}$$

$$x = x + v\Delta t \tag{8}$$

where ω is called the "*inertia weight*" that controls the impact of the previous velocity of the particle on its current one. The parameters c_1 and c_2 are positive constants, called "*acceleration coefficients*". The parameters r_1 and r_2 are random numbers that are uniformly distributed in the interval [0, 1]. These random numbers are updated every time when they occur. The parameter Δt stands for the given time-step. The population of particles is then moved according to (7) and (8), and tends to cluster together from different directions. However, a maximum velocity v_{max}, should not be exceeded by any particle to keep the search within a meaningful solution space [32]. The PSO algorithm runs through these processes iteratively until the termination criterion is satisfied [33, 34].

3.2 Sequential Quadratic Programming

SQP is an iterative method for nonlinear optimization. It is commonly used on twice continuously differentiable problems. It solves the optimization problem by dividing it into a sequence of optimization sub-problems, each which optimizes a quadratic model of the objective subject to a linearization of the constraints [35]. Suppose a nonlinear programming problem of the form as

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } b(x) \geq 0, c(x) = 0 \end{aligned} \tag{9}$$

The Lagrangian for this problem is

$$L(x, \lambda, \sigma) = f(x) - \lambda^T b(x) - \sigma^T c(x) \tag{10}$$

here λ and σ are Lagrange multipliers. At the kth iteration, the current solution point is x_k, the direction is d_k, then the subproblem of SQP is defined as

$$\begin{aligned} &\min_d L(x_k, \lambda_k, \sigma_k) + \\ &\nabla L(x_k, \lambda_k, \sigma_k)^T d + \\ &\frac{1}{2} d^T \nabla^2_{xx} L(x_k, \lambda_k, \sigma_k) d \\ &\text{s.t. } b(x_k) + \nabla b(x_k)^T d \geq 0, \\ &c(x_k) + \nabla c(x_k)^T d = 0 \end{aligned} \tag{11}$$

3.3 PSOSQP

The pseudo code of the PSOSQP applied for image

registration can be concluded as follows.

Step 1 Input two images: the reference image *A* and the input image *B*;
Step 2 Establish the optimization function according to formula (2);
Step 3 Initialization. Generate random particles with random velocities;
Step 4 Evaluate the objective function of each particle;
Step 5 Update the velocity and position of each particle according to formula (7)(8);
Step 6 Order the particles with terms to their objective values;
Step 7 Select the *gbest* particle;
Step 8 Solve the objective function via SQP with *gbest* as the starting point;
Step 9 Replace *gbest* with the final solution obtained by SQP;
Step 10 Jump to Step 4 until the termination criteria is met.

4 Experiments

The experiments were carried out on the platform of Windows XP on desktop PC with Intel Pentium4, 3GHz processor and 2GB RAM The algorithm was in-house developed via the statistics toolbox and fuzzy toolbox of Matlab 2011b. Our program can run at any desktop installing Matlab.

4.1 Convergence Comparison

Suppose n=400, k=4, and d=2. Non-overlapping, partially overlapping, and severely overlapping artificial data were generated randomly from a multivariate Gaussian distribution. The distributions of the data were shown in Fig. 3.

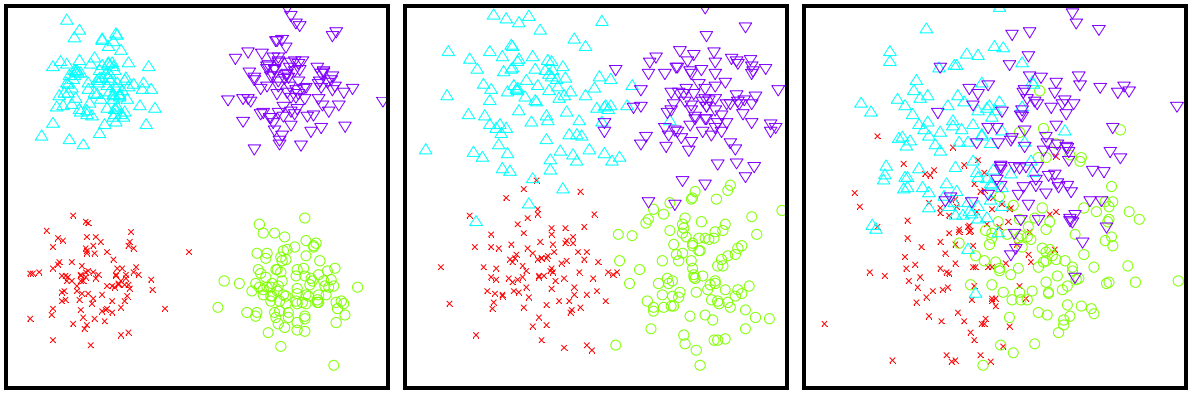

Fig. 3 Artificial Data to three different degrees (a) Non-overlapping; (b) Partially Overlapping; (c) Severely Overlapping

The proposed PSOSQP was tested in comparison with the GA [22] and CPSO [14] algorithm. Each algorithm was run 20 times to reduce the randomness. The results are listed in Table 1. It indicated that for non-overlapping instances, the three algorithms: GA, CPSO, and PSOSQP, can all find the maximal VRC of 1683.2 at least once. The mean VRCs of those algorithms are 1321.3, 1534.6, and 1603.5, respectively. The worst VRCs are 451.0, 1023.9, and 1479.3, respectively. For partially overlapping instances, the three algorithms: GA, CPSO, and PSOSQP, can all find the maximal VRC of 620.5, the mean VRCs of those algorithms are 594.4, 607.9, and 619.5, respectively, and the worst VRCs are 512.8, 574.1, and 591.4, respectively. For severely overlapping instances, the GA, CPSO, and PSOSQP can find the maximal VRC of 275.6, the mean VRCs are 184.1, 203.8, and 219.8, respectively, and the worst VRCs are 129.0, 143.5, and 127.6, respectively.

Table 1 indicates that the mean value of the VRCs obtained by 20 runs of the proposed PSOSQP algorithm is the highest among all three algorithms; therefore, the PSOSQP is the most robust and effective.

Table 1 Experiment results for 400 artificial data sets for 20 different runs

Overlapping Degree	VRC	GA	CPSO	PSOSQP
Non	Best	**1683.2**	**1683.2**	**1683.2**
	Mean	1321.3	1534.6	**1603.5**
	Worst	451.0	1023.9	**1479.3**
Partially	Best	**620.5**	**620.5**	**620.5**
	Mean	594.4	607.9	**619.5**
	Worst	512.8	574.1	**591.4**
Severely	Best	**275.6**	**275.6**	**275.6**
	Mean	184.1	203.8	**219.8**
	Worst	129.0	**143.5**	127.6

4.2 Time Comparison

Moreover, the average time of each algorithm was listed in Table 2. For non-overlapping instances, the average computation time of GA, CPSO, and PSOSQP are 3.34, 3.13, and 2.44 seconds, respectively. For

partially overlapping instances, the average computation time of GA, CPSO, and PSOSQP are 4.10, 3.86, and 3.09 seconds, respectively. For severely overlapping instances, the average computation time of GA, CPSO, and PSOSQP are 4.87, 5.08, and 4.51 seconds, respectively. Table 2 indicates that the average computation time of PSOSQP is the least among all three approaches, besides, the computation time increases as along as the degree of overlapping increases.

Table 2 Average Computation Time (s) of 20 runs

Overlapping Degree	GA	CPSO	PSOSQP
Non-overlapping	3.34	3.13	**2.44**
Partially overlapping	4.10	3.86	**3.09**
Severely overlapping	4.87	5.08	**4.51**

5 Conclusions

In this paper, we first investigate the optimization model including both the encoding strategy and the criterion function of VRC. Afterwards, the PSOSQP algorithm was introduced for solving the model. Experiments on three types of artificial data with different overlapping degrees all demonstrate the PSOSQP is more robust and costs less time than either GA or CPSO.

Future works contains following points: 1) Develop a method that can determine the number of clusters automatically; 2) Use more benchmark data to test the PSOSQP; 3) Apply our PSOSQP to practical clustering problems, including mathematics [36], face estimation [37], image segmentation [38], Image Classification [39], and prediction [40].

References

[1] D. Coomans, C. Smyth, I. Lee, T. Hancock, J. Yang, Unsupervised Data Mining: Introduction, in: D.B. Stephen, Rom, x00E, Tauler, W. Beata (Eds.) Comprehensive Chemometrics, Elsevier, Oxford, 2009, pp. 559-576.

[2] J. D. Bartley, R. W. Buddemeier, D. A. Bennett, Coastline complexity: a parameter for functional classification of coastal environments, Journal of Sea Research, 46(2) (2001) 87-97.

[3] Ivan Kojadinovic, Hierarchical clustering of continuous variables based on the empirical copula process and permutation linkages, Computational Statistics & Data Analysis, 54(1) (2010) 90-108.

[4] Yudong Zhang, Lenan Wu, Bankruptcy Prediction by Genetic Ant Colony Algorithm, Advanced Materials Research, 186 (2011) 459-463.

[5] Yudong Zhang, Lenan Wu, Shuihua Wang, Magnetic Resonance Brain Image Classification by an Improved Artificial Bee Colony Algorithm, Progress in Electromagnetics Research, 116 (2011) 65-79.

[6] Yudong Zhang, Lenan Wu, Crop Classification by forward neural network with adaptive chaotic particle swarm optimization, Sensors, 11(5) (2011) 4721-4743.

[7] Yaozhang Pan, Shuzhi Sam Ge, Abdullah Al Mamun, Weighted locally linear embedding for dimension reduction, Pattern Recognition, 42(5) (2009) 798-811.

[8] Yudong Zhang, Shuihua Wang, Lenan Wu, Spam detection via feature selection and decision tree, Advanced Science Letters, 5 (2012) 726-730.

[9] Yudong Zhang, Lenan Wu, A Rotation Invariant Image Descriptor based on Radon Transform, International Journal of Digital Content Technology and its Applications, 5(4) (2011) 209-217.

[10] Yudong Zhang, Shuihua Wang, Yuankai Huo, Lenan Wu, Feature Extraction of Brain MRI by Stationary Wavelet Transform and its Applications, Journal of Biological Systems, 18(s) (2010) 115-132.

[11] Nicolas Picard, Frédéric Mortier, Vivien Rossi, Sylvie Gourlet-Fleury, Clustering species using a model of population dynamics and aggregation theory, Ecological Modelling, 221(2) (2010) 152-160.

[12] Ivan Kojadinovic, Agglomerative hierarchical clustering of continuous variables based on mutual information, Computational Statistics & Data Analysis, 46(2) (2004) 269-294.

[13] Costel Sârbu, Katharina Zehl, Jürgen W. Einax, Fuzzy divisive hierarchical clustering of soil data using Gustafson-Kessel algorithm, Chemometrics and Intelligent Laboratory Systems, 86(1) (2007) 121-129.

[14] B. Jarboui, M. Cheikh, P. Siarry, A. Rebai, Combinatorial particle swarm optimization (CPSO) for partitional clustering problem, Applied Mathematics and Computation, 192(2) (2007) 337-345.

[15] Shihong Yue, Jeen-Shing Wang, Teresa Wu, Huaxiang Wang, A new separation measure for improving the effectiveness of validity indices, Information Sciences, 180(5) (2010) 748-764.

[16] K. M. Pekpe, S. Lecoeuche, Online clustering of switching models based on a subspace framework, Nonlinear Analysis: Hybrid Systems, 2(3) (2008) 735-749.

[17] Pierre Bellec, Pedro Rosa-Neto, Oliver C. Lyttelton, Habib Benali, Alan C. Evans, Multi-level bootstrap analysis of stable clusters in resting-state fMRI, NeuroImage, 51(3) (2010) 1126-1139.

[18] M. Tamer Ayvaz, Halil Karahan, Mustafa M. Aral, Aquifer parameter and zone structure estimation using kernel-based fuzzy c-means clustering and genetic algorithm, Journal of Hydrology, 343(3-4) (2007) 240-253.

[19] Dong-Xia Chang, Xian-Da Zhang, Chang-Wen Zheng, Dao-Ming Zhang, A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem, Pattern Recognition, 43(4) (2010) 1346-1360.

[20] Carlos Cotta, Pablo Moscato, A memetic-aided approach to hierarchical clustering from distance matrices: application to gene expression clustering and phylogeny, Biosystems, 72(1-2) (2003) 75-97.

[21] Elcio Sabato de Abreu e Silva, Hélio Anderson Duarte, Jadson Cláudio Belchior, An approach based on

genetic algorithms and DFT for studying clusters: (H2O)n (2 n 13) cluster analysis, Chemical Physics, 323(2–3) (2006) 553-562.
[22] Jun-Lin Lin, Meng-Cheng Wei, Genetic algorithm-based clustering approach for k-anonymization, Expert Systems with Applications, 36(6) (2009) 9784-9792.
[23] Dong-Xia Chang, Xian-Da Zhang, Chang-Wen Zheng, A genetic algorithm with gene rearrangement for K-means clustering, Pattern Recognition, 42(7) (2009) 1210-1222.
[24] Bruno Agard, Bernard Penz, A simulated annealing method based on a clustering approach to determine bills of materials for a large product family, International Journal of Production Economics, 117(2) (2009) 389-401.
[25] Taher Niknam, Babak Amiri, An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis, Applied Soft Computing, 10(1) (2010) 183-197.
[26] Roy Gelbard, Abraham Carmeli, Ran M. Bittmann, Simcha Ronen, Cluster analysis using multi-algorithm voting in cross-cultural studies, Expert Systems with Applications, 36(7) (2009) 10438-10446.
[27] Yudong Zhang, Lenan Wu, Rigid Image Registration by PSOSQP Algorithm, Advances in Digital Multimedia, 1(1) (2012) 4-8.
[28] Chia-Ping Shen, Wen-Chung Kao, Yueh-Yiing Yang, Ming-Chai Hsu, Yuan-Ting Wu, Feipei Lai, Detection of cardiac arrhythmia in electrocardiograms using adaptive feature extraction and modified support vector machines, Expert Systems with Applications, 39(9) (2012) 7845-7852.
[29] Yudong Zhang, Lenan Wu, Shuihua Wang, Yuankai Huo, Chaotic Artificial Bee Colony used for Cluster Analysis, Communications in Computer and Information Science, 134(1) (2011) 205-211.
[30] Moreno Marzolla, Fast training of support vector machines on the Cell processor, Neurocomputing, 74(17) (2011) 3700-3707.
[31] Yudong Zhang, Lenan Wu, A Novel Method for Rigid Image Registration based on Firefly Algorithm, International Journal of Research and Reviews in Soft and Intelligent Computing, 2(2) (2012) 141-146.
[32] L. Lamberti, An efficient simulated annealing algorithm for design optimization of truss structures, Computers & Structures, 86(19-20) (2008) 1936-1953.
[33] Aijun Liu, Yu Yang, Hao Yao, Liang Xuedong, Improved Collaborative Particle Swarm Algorithm for Job Shop Scheduling Optimization, Advanced Science Letters, 4(6) (2011) 2180-2183.
[34] Yudong Zhang, Lenan Wu, Bacterial chemotaxis optimization for protein folding model, in: 5th International Conference on Natural Computation, ICNC 2009, August 14, 2009 - August 16, 2009, IEEE Computer Society, Tianjian, China, 2009, pp. 159-162.
[35] José F. Rodríguez, John E. Renaud, Brett A. Wujek, Ravindra V. Tappeta, Trust region model management in multidisciplinary design optimization, Journal of Computational and Applied Mathematics, 124(1-2) (2000) 139-154.
[36] Yudong Zhang, Lenan Wu, Recursive Structure Element Decomposition Using Migration Fitness Scaling Genetic Algorithm, Advances in Swarm Intelligence, 6728 (2011) 514-521.
[37] Yudong Zhang, Lenan Wu, Zhengchao Dong, Shuihua Wang, Face Orientation Estimation by Particle Swarm Optimization, in: Information Science and Engineering (ISISE), 2009 Second International Symposium on, 2009, pp. 388-391.
[38] Yudong Zhang, Lenan Wu, Segment-based Coding of Color Images, Science in China Series F: Information Sciences, 52(6) (2009) 914-925.
[39] Yudong Zhang, Shuihua Wang, Lenan Wu, Yuankai Huo, PSONN used for Remote-Sensing Image Classification, Journal of Computational Information Systems, 6(13) (2010) 4417- 4425.
[40] Yudong Zhang, Lenan Wu, Shuihua Wang, Bacterial foraging optimization based neural network for short-term load forecasting, Journal of Computational Information Systems, 6(7) (2010) 2099-2105.