Multi-objective clustering with Particle Swarm Optimization

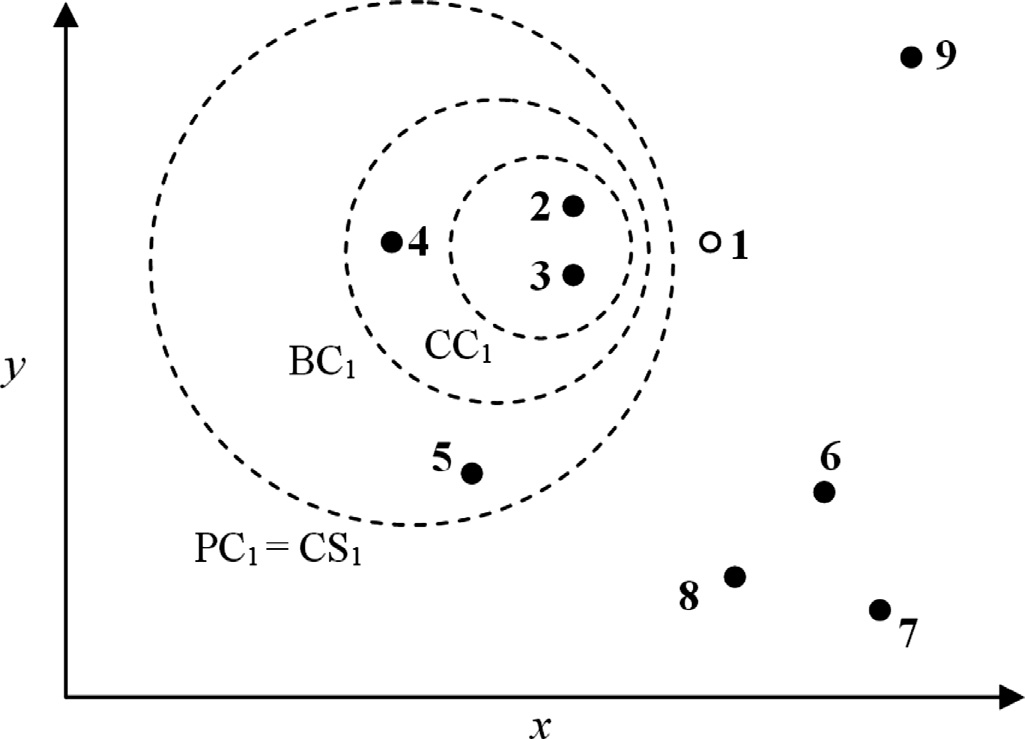
A modified version of multi-objective clustering with Particle Swarm Optimization (MCPSO) algorithm implemented in the study is described in this section. The algorithm optimizes two conflicting objective functions, based on connectivity and cohesion with the aim of obtaining well-separated, compact, and connected clusters. The results of the optimization process is an optimal clustering solution with determined number of clusters.

Objective functions

Optimization criteria needs to be chosen as objectives functions that reflect different aspects of a clustering solution. Two complementary objectives are therefore selected: one based on connectivity, the other one based on similarity of objects in the same cluster.

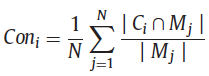
As an objective reflecting cluster connectedness, connectivity measure is utilized which evaluates the degree to which neighboring data points have been placed in the same cluster (Handl & Knowles, 2007). To compute the value of connectivity objective function, a neighborhood has to be constructed first.

A neighbourhood can be defined as a refined group of locally similar data points. To construct these neighborhoods, a neighborhood-based clustering method has to be applied. MCPSO uses a locally adaptive neighbourhood construction algorithm (NC) proposed in Inkaya, Kayalıgil, and Özdemirel (2015), as it’s shown that it performs better in comparison with other traditional neighborhood methods such as *k*-nearest neighbour and *ε*-neighbourhood at dealing with clusters of arbitrary shape and density, as well as outliers (Inkaya & Özdemirel, 2013). NC has also the advantage of being parameter-free, requiring no a priori information about the dataset and producing a neighbourhood unique to each data point. The algorithm determines the density based connectivity and proximity relations among the points, extracted in an adaptive manner using Gabriel graph (GG) (Gabriel & Sokal, 1969). For each data point *i* in *D*, NC algorithm follows four consecutive steps: (1) check the direct connectivity and create the core neighbors set CC*i*; (2) the density tracking check and extraction of the density connected neighbors set BC*i*; (3) the indirect connectivity check and extraction of the extended neighbors set PC*i*; (4) the mutual connectivity tests and selection of final neighbors set CS*i*.

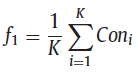


**Fig. 1.** Example dataset for the NC algorithm. CC1, BC1, PC1, and CS1 include the *core neighbors, density connected neighbors, extended neighbors* , and *final neighbors* of point 1, respectively.

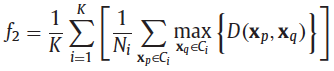
The NC procedure is precomputed only once in the initialization phase of the proposed algorithm. The connectivity of cluster *Ci* with respect to all sub-clusters can then be computed in the optimization phase as:



Connectivity for all *K* clusters is defined therefore as:



Cohesion is another objective, which measures how closely related are objects in a cluster. It can be measured by the sum of the maximum within-cluster distances between data points:



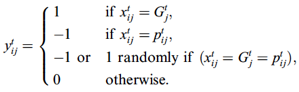
Particle representation

Original MCPSO applies locus-based adjacency genetic scheme proposed in Park and Song (1998) to construct particles in a swarm. The paper doesn’t elaborate however on how the combinatorial form of PSO is combined with the locus-based representation, and no PSO-based clustering algorithm has been found in the literature utilizing this particle representation. Therefore, label-based representation is used instead, which is also used in Jarboui et al. (2007), the paper MCPSO algorithm is based on. Label-based representation consists of an integer vector of *N* positions, where *N* is the number of dataset objects. Each position corresponds to a particular data point and has a value over the alphabet {1, 2, 3, ..., *k*}, where *k* is a number of clusters.

Combinatorial optimization

As it is not possible to use PSO in its original continuous form, within the MO framework,extended version of PSO introduced by Jarboui, Cheikh, Siarry, and Rebai (2007)

is used. Let a X*ti=*{*xti*1*, xti*2,…, *xtin*} be a clustering solution of ith particle at iteration t. Y*ti=*{*yti*1*, yti*2,…,*ytin*} is then a vector, an extra parameter in the combinatorial form of PSO, associated with X, used to permit the transition from the combinatorial state to the continuous state and vice versa. Its jth element is defined as follows:

,

where Gtj and *ptij* are jth elements of the global best vector *Gt=*{*Gt*1, *Gt*2, …, *Gt*n} and the personal best vector *p*t*=*{*pti*1, *pti*2, …, *ptin*}, respectively. Fig. 2 depicts the update rules of the particle:

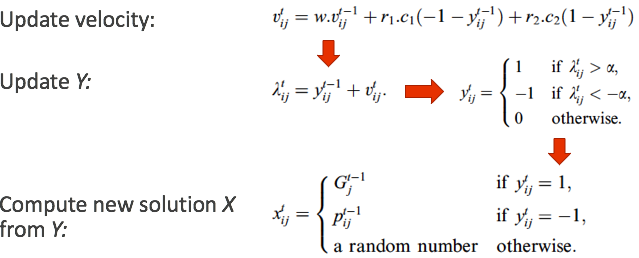


Fig. 2. Particle update rules

Multi-objective PSO

*Initialization*. A random distribution of the initial swarm is generated by using the K-means++ algorithm with different numbers of clusters *k*, and an initial set of random velocities is assigned to them. The personal best for each particle is initialized to the starting location of that particle. The value of *k* is set to vary from 2 to *K*max, where *K*max denotes the maximum possible cluster number existing in the data set and is taken to be √n (n is the number of data objects in the data set), which has been widely used and demonstrated to be suitable in clustering literature (Sheng et al, 2016). The suitability of using such a *K*max value can be further confirmed by the fact that the numbers of clusters in almost all existing data sets, whose structures are known beforehand, are less than their corresponding *K*max values. The value of *K*max can certainly be increased to n, ho if necessary. However, this will result in a higher search space. By employing such a *K*max value, the number of possible k values is existing in population will therefore equal to *K*max − 1, covering the cluster numbers ranging from 2 to *K*max. The swarm size *N* is set to 2\*(*K*max − 1). It can certainly be set 3\*(*K*max − 1) or higher, but will result in a higher time consumption. The set value of *N* ensures that number of *k* is evenly spread across the population at the beginning of the optimization process. It’s been concluded from the experiments that this configuration results in a good final clustering solution more consistently than randomly generated *k* value for K-Means.

*Analysis*. The concept of MaxiMin strategy is applied in the original MCPSO algorithm to determine pareto optimal solutions. It has been substituted here with the Weighted Sum method, as it performs better than MaxiMin strategy in the context of modified algorithm implemented in this study, which might be due to the change of particle representation. Given clustering solution xi of ith particle in the swarm and its corresponding values of objective functions *Fi* = { *f1(xi)*, *f2(xi)*,…, *fn(xi)*}, the implemented weighted sum fitness function is computed using the distances of the objective values from an ideal solution, called utopia point. The fitness is thus defined as follows:

,

where *wj* is the weight of jth distance, *f \*j* is the best coordinate of utopia point obtained for jth objective function during the optimization process. The value of *p* is set to 2. As it can be inferred from the formula, the lower value of the fitness indicates a better clustering solution.

*Personal best selection*. If the fitness value of the current position of particle *i*, xk+1i is lower than the fitness value of the previous personal best position, pki, at iteration *k*+1, then pk+1i is set to the particle’s current position.

*Leader selection.* A leader is selected from the top portion of the best particles at each iteration.

*Updating velocity vector and particles’ positions*. The process of updating the particles’ positions is depicted on Fig. 2. The inertia weight factor at iteration *k*, *wk*, is dynamically adjusted throughout the optimization process:

,

where *k*max is the maximum number of iterations.

*Termination criterion*. A maximum number of iteration *k*max is applied to terminate the algorithm. In addition, if the best clustering solution in the swarm doesn’t change for 50 consecutive iterations, the algorithm is terminated.

Parameter settings

A maximum number of generations *k*max is set to 200 to reduce the running time of the algorithm. The recommended in the original paper values of cognitive and social parameters, *c*1 = 1*.*42 and *c*2 = 1*.*63, are used. As suggested, the maximum value for the inertia weight factor, *w*max is set to 0.9, while the minimum value *w*min - to 0.4. To keep the balance between intensification and diversification of the algorithm, decision parameter 𝜶 is set to 0.5. A time stamp unit Δ*t* = 1 is used. All parameter settings used in the implemented algorithm are summarized in Table 1.

Table 1. Parameter of implemented MCPSO algorithm.

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Value |
| *Δt* | Time step value | 1 |
| *c*1 | Cognitive parameter | 1.42 |
| c2 | Social parameter | 1.63 |
| *r*1 | Random value | [0, 1] |
| *r*2 | Random value | [0, 1] |
| *w*max | Maximum value of the inertia weight factor | 0.9 |
| *w*min | Minimum value of the inertia weight factor | 0.4 |
| 𝜶 | Decision maker | 0.5 |
| *k*max | Maximum number of iterations | 200 |
| *K*max | Maximum number of clusters | *√n* |
| *N* | Number of particles | 2 \* *K*max |

References

J. Handl and J. Knowles, “An Evolutionary Approach to Multiobjective Clustering,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 56–76, Feb. 2007.

K.R.Gabriel,R.R.Sokal,New statistical approach to geographic variation analysis, Syst. Zool.18 (3) (1969) 259–278.

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

W. Sheng, S. Chen, M. Sheng, G. Xiao, J. Mao, and Y. Zheng, “Adaptive Multisubpopulation Competition and Multiniche Crowding-Based Memetic Algorithm for Automatic Data Clustering,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 838–858, Dec. 2016.