Multi-objective Clustering Algorithm Using Particle Swarm Optimization with Crowding Distance (MCPSO-CD)

Alwatben Batoul Rashed A[#] [*], Hazlina Hamdan[#], Nurfadhlina Mohd Sharef, Md Nasir Sulaiman,
Razali Yaakob, Mansir Abubakar
[#]University Putra Malaysia. Malaysia
43400 UPM Serdang Selangor Darul Ehsan
hazlina@upm.edu.my

[*]Qassim University, P.O.Box 6633, Qassim 14452, Sudia Arabia

ABSTRACT: *Clustering as an unsupervised method is used as a solution technique in various fields to divide and restructure data to become more significant and to transform them into useful information. Currently, clustering is being a difficult problem and complex phenomena since an appropriate number of clusters is unknown, the large number of potential solutions, and the dataset being unsupervised. The problems can be addressed by Multi-objective Particle Swarm Optimization (MOPSO). In Knowledge Discovery settings, complex optimization problems are globally explored with Particle Swarm Optimization (PSO). Lack of appropriate leader selection method becomes a serious issue associated with PSO techniques. In an attempt to address this problem, we proposed a clustering-based method that utilizes the crowding distance (CD) technique to balance the optimality of the objectives in Pareto optimal solution search. We evaluated our method against five clustering approaches that have succeeded in optimization, these are: The K-means Clustering, the IMCPSO, the Spectral clustering, the Birch, and the average-link algorithms. The results of the evaluation show that our approach exemplifies the state-of-the-art methods with significance difference in all most all the tested datasets.*

Keywords: Knowledge Discovery, Data Clustering, Crowding Distance, Particle Swarm Optimization, K-means Clustering

DOI: 10.6025/jisr/2020/11/1/21-30

Received: 2 October 2019, Revised 22 December 2019, Accepted 9 January 2020

1. Introduction

Discovering useful information from the collection of data is known as knowledge discovery. This process is carried out using many techniques and methods which includes classification, clustering, regression and summarization [1]. Data clustering is a technically powerful task in an area of data mining and knowledge discovery. It involves partitioning the datasets into a chunk of clusters of similar features [2].

K-means [3] and Hierarchical [4] clustering algorithms are widely used and appeared to be popular among the research communities and industries as well. K-means algorithm partitions the datasets into a set of k groups of similar entities with predefined k value [5], while the hierarchical algorithm builds a tree of clusters. In addition to K-means and Hierarchical there are many clustering algorithms that performed well in datasets clustering such as BIRCH and average-link. BIRCH is a scalable clustering algorithm regarding the number of objects, it displays good quality in clustering datasets [6], the average linkage is a hierarchical clustering technique where the distance between two clusters is defined as the average distance in a cluster between each point. All these algorithms concentrated on a single objective optimization while in some real–world problems multiple optimizations of objective functions are required [7]. Furthermore, some of these classical algorithms have issues in initial centroid selection, they have a low rate in convergence [1] and trapped in local optima.

Swarm-based clustering algorithms have been successful in solving clustering issues [8], [9], [10], [11], [12]. Particle Swarm optimization is developed to have a random distribution nature. It is a simple method for searching the approximate optimal solutions and requires no much effort in parameter configuration. Complex optimization problems are globally explored with PSO. The ability of a Multi-objective particle Swarm Optimization (MPSO) algorithm to solve a clustering problem as applied in the population of a solution space allows all the Pareto set to be approximated in one run. To solve the issues associated with clustering problems, several multi-objective clustering algorithms are proposed in the literature [13], [14],[15]. However, some loopholes identified in the recent works include that in a large dataset the curse of data dimensionality may be encountered because the expected clusters are usually a combinatorial problem. Likewise, the selection of global leader may be tedious due to the unique feature in clustering problem on the Pareto set distribution. These major issues are a serious threat to the realization of clustering in MPSO in terms of practical performance of the algorithm. To address these issues, Abubakar et al. [14] propose a hybrid multi-objective clustering approach based on PSO and Simulated Annealing with the aim of estimating the number of clusters and split them into without knowing the actual number of such clusters. The approach uses artificial datasets to study the efficiency of their algorithm based on the parameters of the particles' velocity. Furthermore, Amano et al. [15] proposed a multi-object particle swarm optimization technique that defines a partition-based clustering as a multi-objective problem. The goal of the technique is to obtain is to obtain a well separated, connected, compact clusters. The technique uses two objective functions defined based on the concept of connectivity and cohesion. Inter and intra cluster relationships among the particles were missing in the implementation. In addition, Gong et al. [16] proposed a multi-objective clustering framework that makes use of Particle Swarm Optimization. The method is named "Improved Multi-objective Clustering Particle Swarm Optimization (IMCPSO)" framework. Even though, the approach provides an improvement in the performance as suggested but demonstrated a setback in clustering distributions solutions that have a negative effect on the performance of leaders' selection thereby making the optimization models fall into local optima rather than the global optima.

One important step in MOPSO algorithm is the selection of leaders which affects the convergence ability of the algorithm as well as the preservation of the extensions for non-dominating solutions. Consequently, this research adopted the technique used in [16] with some changes in leader selection method. In addition, this study proposed a crowding distance (CD) technique to balance the optimality of the objectives in search of the Pareto Optimal solution. Moreover, it applies the same MOPSO approach with crowding distance algorithm proposed in [17], but employs a different technique to respective size of the set of leaders. Such that best leaders are maintained with respect to their crowding distance when the size of the set of leaders is greater than the maximum permissible size. Hence, this improvement in the size of the leaders' set chosen based on the dominated concept to guarantee the survival of the best solution. According to this strategy, not only the crowding value became affected by the leader's selection but also the dominated concept.

2. Literature Review

2.1. Multi-objective Optimization and Swarm Intelligence

In order to treat optimization problems that have complex nature, Swarm Intelligence (SI) which is the family of methods with decentralized and Self-Organisation properties of artificial or natural activities. Different algorithms have been in place for solving problems of different form applied in several application domains [18]. Even though these algorithms demonstrated high performance, they are simply behind single objective optimizations. Hence, the need for a population-based and robust multi-objective optimization approaches to address multi-objective problems.

Multi-objective problems are problems with more than one objective. Usually these objectives used to conflict with one another. The distinction of multi-objective optimization algorithms and a single - objective optimization algorithms is that in multi - objective algorithms problems, multiple objectives are considered for computing the optimal solutions, as the name suggests. While, in single-objective problems, there is always a single optimum solution [19][20]. The implementation of multi-objective is based on *EA*. The basic concept of a multi-objective minimization problem is defined below:

2.2. Basic Concept

Minimization of a multi-objective problem can be defined as followed:

$$\text{Minimize}\, f(x) = [f_1(x),...f_k(x)] \qquad (1)$$

Subject to the constraints:

$$g_i(x) \leq 0,\ i = 1, 2, \ldots\ldots m \qquad (2)$$

$$h_i(x) \leq 0,\ i = 1, 2, \ldots\ldots p \qquad (3)$$

Where,

$x = (x_1, x_2, ... x_n) \in$ population space $g_i\ (x)$ while the constraint function of the problem is defined by $h_i\ (x)$. The number of objectives is denoted by k, and the number of constraints for both equality and inequality are represented by m and p respectively.

Multi-objective optimization leads to an optimal solutions' set known as non-nondominated solutions. The Dominated concept can be described a solution vector x said to be *Pareto dominate* (represented as $X < Y$) if and only if X is strictly better than Y in at least one objective, and X is no worse than Y in any of the considered objectives. It is called *Pareto- optimal* if it is not subjected by any other solution of an existing population. *The Pareto-optimal front* is regarded as the collection of all the *non-dominated* solutions in the objective space [23].

MOEAs are mainly appropriate for multi-objective solutions. They search for manifold solutions that are optimal in a parallel way. MOEAs can find a set of solutions in a single run for its final subjects. Out of the set of solutions, the most suitable solution would be chosen via the criterion preference. Thus, the primary aim of a multi-objective search algorithm is to determine the set of solutions that can best fit the approximation of a Pareto front. However, the serious challenge associated with multi-objective search algorithms is the accuracy-interpretability trade-off [25], [26], [20], [21].

2.3. Multi-objective PSO

Particles are presented as the potential solutions in PSO framework and achieved global optimization via the shift of its location in *D*-dimension search space. Each of the particle in the PSO framework is regarded to be a possible solution, and the particles attained optima by shifting their positions in a d-dimension order within the search space. The following justifies the velocity vi and position *xi*:

$$v_i(t+1) = v_i(t) + c_1 r_1 (pbest_i - x_i) + c_2 r_2 (gbest - x_i) \qquad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t) \qquad (2)$$

Where,

$\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{iD})$ represents particle in the i^{th} position, while the velocity of particle i is given by $\boldsymbol{v_i} = (v_{i1}, v_{i2}, \ldots, v_{iD})$, *t is the number of generation*, while the learning factors, local and global are represents by c_1 and c_2, and the random order ranging between [0, 1] are represented by r_1 and $r_{2,}$ respectively.

pbest$_i$ is the particle i previous best position, and

gbest is the overall (global) best position of the particles found within entire swarm. Equation (2) is use for updating the particle position in the swarm.

The algorithm can easily control the search space and produces the best results in some certain problems by introducing an inertia weight factor w [31], the velocity of the particle is then adjusted to

$$v_i(t+1) = w.v_i(t) + c_1 r_1 (p_i - x_i) + c_2 r_2 (g - x_i) \qquad (3)$$

MOPSO seemed to be distinct from PSO because it includes the processes of constructing and preserving the external archive using the Pareto optimality concept. Furthermore, the machinery in terms of the selection is quite different on gbest and pbest. Consequently, the latest multi-objective PSO works with the optimal Pareto concept to select non-dominated particles as the leaders in order to converge solutions to the optimal Pareto set values. [22] and [23].

2.4. Clustering Problem

Clustering problem can be considered as dataset $P = \{ p_1, p_2,..., p_n\}$, where $pi = (p_{i1}, p_{i2},..., p_{id})$ is a d- dimensional feature vector that referred to the object, p_{ij} is the object's feature value I in dimension j while n referred to the quantity of objects in P. P clustering is dividing P into k clusters $\{ C_1, C_2,... ,C_k\}$ with the following f properties: P clustering is dividing P into k clusters $\{C_1, C_2,\ldots,C_k\}$ both with the f properties as follows:

(1) $U_{i=1}^{k} c_i = P$,

(2) $C_i \cap C_j \neq \emptyset$, such that $i, j \in \{1, 2,..., k\}$ and $i \neq j$.

3. Proposed Technique

In this section a proposed Multi-objective Clustering Particle Swarm Optimization with crowding distance (MCPSO-CD) method is described. The method is grounded on particle swarm optimization model proposed by [18]for a multi-objective situation. This technique consists of optimization level and decision-making level that is designed for clustering purpose. The former provides an optimal solution for a given clustering problem known as Pareto solutions and each of the solutions is grouped with a different sum of clustered in embedded form. MCPSO-CD uses these solutions to automatically determine the optimal clustered groups. The best solution among the solutions is selected by a simple decision which is the function of a later level. This is also the case in any Pareto solution to be considered optimal.

3.1. Objective Functions

Assuming a clustering solution of a data, different measures of estimation occurs, the target is to locate and select the well separated and dense clusters. Two objective functions are used for that target. First is the overall deviation and second is the mean space of inter-clusters which can be used to evaluate inter and intra cluster separation between the clustered groups.

3.1.1. Overall Deviation

This function determines the general deviation in the intra-cluster size of data used. The overall deviation must be minimized. The formulation is given as:

$$Dev(C) = \Sigma_{Ck} \in C \Sigma_{i \in Ck} \delta(i, \mu_k) \quad (4)$$

Where,

C is a cluster set,

i is a data element,

μ_k is a cluster centroid

C_k, δ (.,.) is a distance function like Euclidean distance.

3.1.2. Clusters mean Distance

An objective function that determines the inter-cluster variation between clustered groups is referred to as "Mean distance between the clusters". It is computed by the minimum space of the neighboring clusters. The N*eighbor* serves as a local model that emulates the relationships between two points of data. *Gabriel graph* is applied to identify in all the data points adjacently [24]. This graph is a sub-graph of the "*Delaunay triangulation"*, that joins the two data nodes v_i and v_j in which no remaining node v_k is inside the open sphere with diameter $[v_i v_j]$. Gabriel graph has merit that it can acquire all the attached graphs with distance that is suitable for the computation. This objective function is given as:

$$Mdc(C) = \frac{1}{|c|} \Sigma_{C_k} \in C \ (min_{i \in C_k, j \in N_i, j \in C_k}, \delta\ (i,j)) \quad (5)$$

Where,

N_i represents the neighbors' data set *i* data in Gabriel graph. *Mdc* must be a maximized objective function. In order to reduce the objective to be the same as *Dev*, *Mdc is the objective function values.*

3.2. Particle Iteration

As stated earlier, MOPSO method can easily be affected by the curse of data dimensionality as the dataset increases. Due to this, MOPSO approach can only provide irregular clustering dispersity in a process of searching and it did not consider previous knowledge in the search process. To overcome the challenge, some properties of clustering (like Gabriel graph and agglomerative clustering [25] are applied to improve and obtain better solutions [16].

Step 1: Determines the topological centers that are in the partition in Gabriel graph. It computes a *c* value of every partition in a clustered solution. Some data whose value is very big is said to be a topological midpoint. When there is more than the center, it will randomly select the topological center.

Step 2: After selecting the topological midpoint, agglomerative clustering method applied. It begins with *k* clusters, where each cluster contains a topology midpoint and other points in the data required to be reassigned. The succession merging continuous until the entire data points are clustered.

Step 3: The final step transformed the improved clusters into a new vector particle; *k* is placed in the first vector element and then group the number correspondence to the data points by dividing it by *k* and put them into another *N* elements.

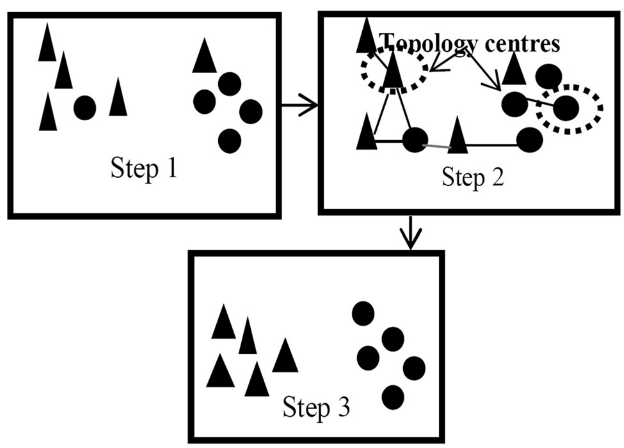

Figure 1. Particle iteration process

3.3. Leader Selection Strategy

A PSO model with Pareto optimum scheme can generate a set of non-dominant leader solutions. But the appropriate selection of a leader becomes an issue since non-dominated solutions are mathematically good. The selection pbest is simple as it adopts non-dominated particle between the previous and current positions as a pbest. Crowding Distance is used in this work for *pbest* selection.

3.3.1. Crowding Distance Computation

It is obviously clear that the main issue when dealing The main issue when dealing with multi-objective PSO is the manner to generalize the leader's concept in the presence of multiple (equally good) solutions. The simplest approach is to consider any non-dominated solution as a new leader. However, this approach has the disadvantage of rapidly increasing the size of leaders. The present study uses the crowding factor technique used in [13] with minor improvements in respects to size of leaders set to guarantee survival of the best solution based on the dominated concept. Crowding distance is measured by first sorting the computed objective function values of the set of solutions in ascending order. A particular value of a crowding distance solution is considered to be the average distance between its two neighboring solutions. Infinite crowding distance values are assigned to the boundary solutions that have the lowest and highest values of the objective function such that they will always be selected. This process is performed across each objective function. The approach then selects a leader for each swarm of particles according to the crowding value of the leaders. The maximum size of the leaders in the set equals to the size of the swarm (or population). The set of leaders is updated after each generation, so are the corresponding crowding values. If the size of the leaders in the set is greater than the maximum permissible size or crowds in some region, the best leaders are retained on the basis of their crowding values. The rest of the leaders are eliminated on the basis of dominated concept.

Algorithm for the computation of crowding distance is shown in Figure 2. Figures 3, 4 and 5 shows the original data, Pareto set with crowding distance and Pareto set with improvement in crowding distance, respectively.

```
1. Get the number of non-dominated solutions in the
   external repository
2. n = | S |
3. Initialize distance
4. FOR i=0 TO MAX
     a. S[i].CDC = 0
     b. Compute the crowding distance of each solution
5. For each objective obj
     a. Sort using each objective value S = sort(S, obj)
6. For i=1 to (n-1)
     a. S[i].CDC = S[i].CDC + (S[i+1].obj - S[i-1].obj)
     b. Set the maximum distance to the boundary points
        so that   they are always selected S[0].CDC =
        S[n].CDC = maximum distance
     c. Delete all solution in set of leaders which are
        dominated
7. For i=1 to (n-1)
     a. S[m1]= S[i].obj
     b. if  (S[i+1].obj < S[m1] ))
          i. delete S[i+1]
     c. else
          i. [m1]= S[i+1].obj
```

Figure 2. Pseudocode for computing crowding distance

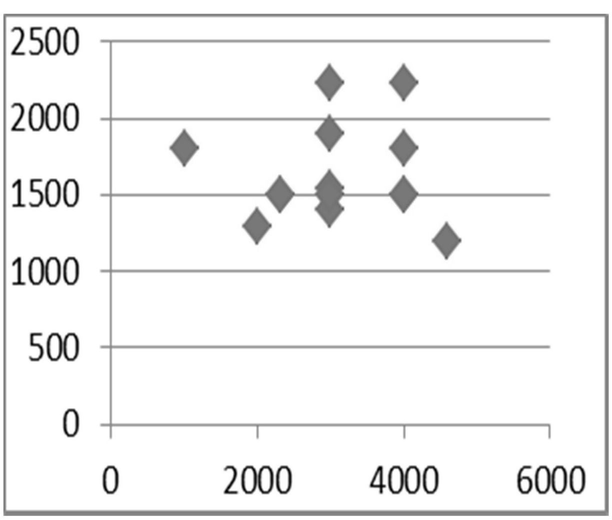

Figure 3. The original data

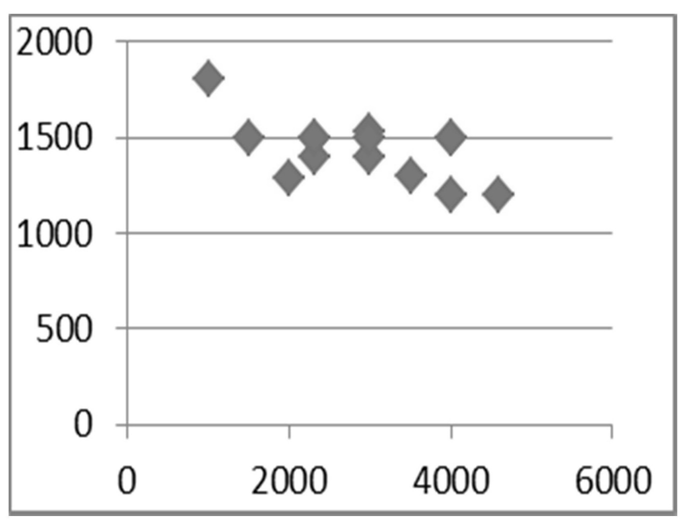

Figure 4. Pareto set with Crowding distance

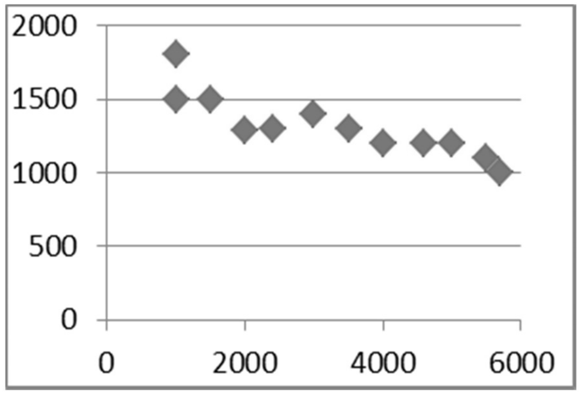

Figure 5. Pareto set with improvement in crowding distance

In figure 4, it can be observed that the Pareto set crowds in some region unlike in fig 5, which shows the improvement in crowding distance technique of the approach proposed in [17] after adding the dominated concept. In the proposed crowding distance, it is not only the crowding value that influences the leader's selection but also the dominated concept to guarantee the survival of the best Pareto set.

3.4. Decision Making

Due to the trade-off among the objective functions, decision making in a residual Pareto set is tedious. However, the sparse coefficient is used to select the particles out of the residue Pareto set. We defined the sparse coefficient to be equation (6) when entire Pareto solutions are normalized.

MCPSO-CD Algorithm

```
Input: itermax set, learning parameters (c1 and c2), Np
Output: Let NonDom be non-dominated set of repositions
Compute the relationship matrix and Gabriel graph of a dataset
Initialize swarm
for each particle do;
 Improve particle in swarm;// Particle Iteration
Evaluate objective functions for particle;
        end for
Store non-dominated particles in NonDom;
 for iter in 1 to itermax do;
         Select gbest in NonDom;  //  Leader selection strategy
          Update particle, pbest, flight;
           for each paricle in swarm do;
                  Improve particle in swarm;// Particle Iteration
Evaluate objective functions for particle;
       end for
 end for
Store non-dominated particles in NonDom;
end for
Select solution in NonDom// Decision making
```

Figure 6. An illustration of MCPSO-CD algorithm

$$sc_i = (d_{l,\,i} + d_{r,\,i}) / 2 \qquad (6)$$

Where sc_i is the sparse coefficient of i^{th} Pareto set, and $d_{l,\,i}$, $d_{r,\,i}$ represents the Euclidian distance that is nearer to both the right part of the Pareto set and left part of the Pareto solution respectively. A solution to be considered is the Pareto solution with a large sparse coefficient. An illustration of MCPSO-CD algorithm is given below:

4. Datasets

In order to obtain an optimum result in performance analyses of our proposed method, we evaluate the technique with seven datasets; five datasets are artificially generated data sets as well as two real-world data sets sourced in KEEL Repository, well-known and being used as benchmark data globally. These datasets have been used in solving particle swarm optimization problems as presented in many scientific articles[16] [26]. Table 1 summarizes the datasets and their properties.

Dataset	***D***	***K***	***N***
Flame	2	2	240
Jain	2	2	373
Path-based	2	3	299
Compound	2	6	399
R15	2	15	600
Glass	9	7	214
House-votes	16	2	232

Table 1. Data Sets Properties

D = The dimension of the data,

K = The number of classes in the data, and

N = The number of data instances.

5. Experimental Setup

We configured the parameters of IMPSO and the configuration results of each the datasets are recorded and analysed. Adjusted Rand Index (ARI) [24] is used to measure the accuracy of our method since there is a standard label in the dataset. The similarity between the generated clusters and true clusters is measured by ARI. When the generated clusters and true clusters have a high degree of similarity, the index will have a high value. We compare our method with the well-known IMCPSO, *K*-means, Spectral clustering, Birch and the Average-link. The parameter values used in the experiment are: $w = 0.85$, $c1$ *and* $c2 = 0.7$, $itermax = 500$, $Np = 20$ *and* $kmax = 15$.

6. Results and Discussion

The experimental results of our proposed method, K-means, IMCPSO, spectral clustering, Birch, and average-link algorithms are recorded. From the results presented in Table 2; it is clearly shown that our method outperforms state-of-the-art techniques in clustering performance. As reported in the result table, the baseline methods outperform the MCPSO-CD in terms of ARI index in one dataset (R15) only. it is worth mentioning that clusters number of (R15) is 15, but the kmax of MCPSO-CD is 15, this is the reason why the MCPSO-CD shows bad performance on (R15) dataset. Furthermore, the performance of MCPSO-CD is better than the baseline clustering methods in one real-world dataset, Glass and remains competitive in the other real-world dataset, House-votes.

In summary, the performance of MCPSO-CD is outperforms to the other clustering techniques in the shape datasets and real-world by improvement in leader selection strategy for MOPSO to avoid models fall into local optima rather than the global optima.

Datasets Type	Datasets	MCPSO-CD	IMCPSO	K-means	Spectral Clustering	Average-link	Birch
Shape cluster datasets	Flame	**0.94**	0.92	0.48	0.26	0.01	0.49
	Jain	**0.93**	0.72	0.51	0.72	0.02	0.92
	Path-based	**0.90**	0.89	0.48	0.50	0.39	0.47
	Compound	**0.87**	0.82	0.50	0.49	0.58	0.78
	R15	0.70	**0.96**	0.19	0.24	0.94	0.93
real-world datasets	Glass	**0.35**	0.31	0.24	0.17	0.02	0.26
	House-votes	**0.64**	0.33	0.61	**0.64**	0.01	0.61

Table 2. Results Evaluated Against Other Techniques

7. Conclusion

Multi-objective PSOs is recommended to effectively solve the optimization problem. In this work, we proposed a possible solution to the clustering problem due to the appropriate number of clusters is unknown. The number of clusters must all be defined to solve this problem and the objects for all of these clusters have to be judiciously assigned. The problems can be addressed by Multi-objective Particle Swarm Optimization (MOPSO) with Crowding Distance. Based on result crowding distance it is an ideal technique in leader selection technique. The result recorded significant improvement with our method compared to the baseline approaches in the result. Even though, our technique remains competitive in some test cases; it does not signify low performance in our approach since the average accuracy in our method outperforms the average accuracy in the baseline techniques in almost all the affected cases.

References

[1] Han, J. (2015). Data Mining/ : Concepts and Techniques, 2015.

[2] Berkhin, P. Survey of Clustering Data Mining Techniques, p 1–56.

[3] Jain, A. K. (2009). Data clustering/ : 50 years beyond K-means q, *PATTERN Recognit. Lett.*, 2009.

[4]Song, W., Di Matteo, T., and Aste, T. (2012). Hierarchical Information Clustering by Means of Topologically Embedded Graphs, 7 (3), p 1–14, 2012.

[5] F. Chen, C. Y., & Ye. (2004). Particle swarm optimization algorithm and its application to clustering analysis, *In*: *Proceedings IEEE Int. Conf. networking, Sens. Control*, 2 (1), p 789–794, 2004.

[6] Mining, D., and Discovery, K. (1997). BIRCH/ : *A New Data Clustering Algorithm and Its Applications*, 182, p 141–182, 1997.

[7] *Soft Computing for Knowledge Discovery and Data Mining*. .

[8] *Data Clustering Algorithms Based On Swarm Intelligence,* p 407–411, 1992.

[9] Chuang, L., Hsiao, C., and Yang, C. (2011). Expert Systems with Applications Chaotic particle swarm optimization for data clustering, *Expert Syst. Appl.*, 38 (12), p 14555–14563, 2011.

[10] Grosan, C., Abraham, A., and Chis, M. (2006). *Swarm Intelligence in Data Mining*, Volume 20, 2006, p 1–20.

[11] Paoli, A., Melgani, F., Member, S., Pasolli, E., Member, S. (2009). Clustering of Hyperspectral Images Based on Multiobjective Particle Swarm Optimization, December, 2009.

[12] Al Moubayed, N., Petrovski, A., Mccall, J. (2011). Clustering-Based Leaders ' Selection in Multi-Objective Particle Swarm Optimisation Clustering-Based Leaders Selection in Multi-Objective Particle Swarm Optimisation, September, 2011.

[13] Yang, F., Sun, T., Zhang, C. (2009). Expert Systems with Applications An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization, *Expert Syst. Appl.*, 36 (6) 9847–9852, 2009.

[14] Abubaker, A., Baharum, A., Alrefaei, M. (2015). Automatic Clustering Using Multi-objective Particle Swarm and Simulated Annealing, 2015.

[15] Armano, G., Farmani, M. R. (2016). *Multiobjective clustering analysis using particle swarm optimization*, volume 55, p. 184–193, 2016.

[16] Gong, C., Chen, H., He, W., Zhang, Z. (2017). *Improved multi-objective clustering algorithm using particle swarm optimization*, p. 1–19, 2017.

[17] Raquel, C. R. (2005). *An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization,* p 257–264, 2005.

[18] Kennedy, J., Eberhart, R. C., Shi, Y. (2001). Swarm Intelligence, *Evol. Comput.*, 78 (2) 512, 2001.

[19] Justesen, P. D. (2009). Multi-objective Optimization using Evolutionary Algorithms Progress report by, 2009.

[20] Konak, A., Coit, D. W., Smith, A. E. (2006). Multi-objective optimization using genetic algorithms/ : A tutorial, volume 91, p 992–1007, 2006.

[21] Cort, A., Ernesto, L., Espinosa, M. (2014). An Experimental Comparison of Multiobjective Algorithms/ : NSGA-II and An Experimental Comparison of MultiObjective Algorithms/ : NSGA-II and OMOPSO, November, 2014.

[22] Santana, R. A., Pontes, M. R., Bastos-Filho, C. J. A. (2009). A multiple objective particle swarm optimization approach using crowding distance and roulette wheel, *ISDA 2009 - 9th Int. Conf. Intell. Syst. Des. Appl.*, June, p. 237–242, 2009.

[23] A. T. Engineering and E. Conversion "01 SOLID FUELS, No. July, p 328–335, 2017.

[24] Kayal, S., Evin, N. (2015). *An adaptive neighbourhood construction algorithm based on density and connectivity*, Volume 52, p 17–24, 2015.

[25] Choo, J., Science, C., Computer, C. C., Ulvi, O., Computer, C. MOSAIC: Agglomerative Clustering with Gabriel Graphs.

[26] Armano, M. R., Farmani, G. (2016). Multiobjective clustering analysis using particle swarm optimization., Number 5, p. 184–193, 2016.