Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Clustering a Big Mobility Dataset Using an Automatic Swarm Intelligence-Based Clustering Method

I. Behravan[1], S.H. Zahiri[1,], S.M. Razavi[1], R. Trasarti[2]*

[1]*Department of Electrical Engineering, Faculty of Engineering, University of Birjand, Birjand, Iran.*

[2]*KDD Lab., ISTI-CNR, Pisa, Italy.*

Article Info

Article History:
Received 10 July 2017
Revised 14 February 2018
Accepted 21 May 2018

Keywords:
Big data clustering
Mobility dataset
K-means
Swarm intelligence
Particle swarm optimization

[*]Corresponding Author's Email Address: hzahiri@birjand.ac.ir

Extended Abstract

Background and Objectives: Big data referred to huge datasets with high number of objects and high number of dimensions. Mining and extracting big datasets is beyond the capability of conventional data mining algorithms including clustering algorithms, classification algorithms, feature selection methods and etc.

Methods: Clustering, which is the process of dividing the data points of a dataset into different groups (clusters) based on their similarities and dissimilarities, is an unsupervised learning method which discovers useful information and hidden patterns from raw data. In this research a new clustering method for big datasets is introduced based on Particle Swarm Optimization (PSO) algorithm. The proposed method is a two-stage algorithm which first searches the solution space for proper number of clusters and then searches to find the position of the centroids.

Results: the performance of the proposed method is evaluated on 13 synthetic datasets. Also its performance is compared to X-means through calculating two evaluation metrics: Rand index and NMI index. The results demonstrate the superiority of the proposed method over X-means for all of the synthetic datasets. Furthermore, a biological microarray dataset is used to evaluate the proposed method deeper. Finally, 2 real big mobility datasets, including the trajectories traveled by several cars in the city of Pisa, are analyzed using the proposed clustering method. The first dataset includes the trajectories recorded in Sunday and the second one contains the trajectories recorded in Monday during 5 weeks. The achieved results showed that people choose more diverse destinations in Sunday although it has fewer trajectories.

Conclusion: Finding the number of clusters is a big challenge especially fir big datasets. The results achieved for the proposed method showed its fabulous performance in detecting the number of clusters for high dimensional and massive datasets. Also, the results demonstrate the power and effectiveness of the swarm intelligence methods in solving hard and complex optimization problems.

Introduction

The huge amount of data created constantly with increasing rate from different sources such as smart phones, social media and imaging technologies becomes difficult to be analyzed by conventional data analytic tools.

For this reason, a new field of research called big data analytics [1] is growing faster in the research and industrial communities. Big data is defined as the dataset whose size is beyond the classical data management tools and processing techniques. In other words, all data life phases must be reconsidered such as the storage, the management and the analysis [2]. Big data analytics, which has attracted more and more attention among researchers, is to automatically extract knowledge from large amounts of data. It can be seen as mining or processing of massive datasets [3]-[5]. There are several challenges in analyzing massive datasets such as large volume of data, dynamical changes of data, data noise and etc. These challenges cause difficulties in extracting hidden patterns and useful information from big data, so new and efficient algorithms should be designed to handle big data analytic problems. One of the most important tools in data mining is clustering technique, in fact it is used in different fields such as, biology, ecology, social science, marketing and psychology as useful tool for pattern recognition and profiling [6]-[9]. Actually the aim of clustering is, to divide the objects of a dataset to a specified number of groups (or clusters) so that the objects within clusters have the most similarity and the most dissimilarity with objects of other clusters. Many techniques have been introduced for clustering such as K-means [10] and Fuzzy c-means [11]. These traditional clustering techniques, fail to give accurate results when dealing with huge amount of data because of their complexity and computational costs. For example, the traditional K-means clustering is NP-hard even when the number of clusters is k = 2. Consequently, scalability is the main challenge for big data clustering [12]. Many real-world applications can be formulized as optimization problems that need kinds of algorithms capable of solving such optimization problems [13]. Most traditional optimization methods are only able to solve non-complex and continuous problems [14]. So, heuristic algorithms are proposed to solve complex and also discrete optimization problems which cannot be solved by the traditional optimization methods.

Recently, swarm intelligence (SI) and evolutionary algorithms (EA), two kinds of heuristics, are attracting more and more attentions from researchers. Swarm intelligence and evolutionary algorithms are collections of population-based methods of searching techniques. To search the solution space of the problem being optimized, these methods hire a population of individuals. Each individual represents a potential solution to the problem. Evolutionary algorithms are inspired by the biological evolution theory [15]. These algorithms use ideas of biological evolution such as reproduction, mutation and recombination for searching the solution space of an optimization problem. Also, they use the Darwin's survival theory for producing new generations of possible solutions and finally finding the best solution among all of the generations. The procedure of the evolutionary algorithms is shown in Fig. 1. Genetic algorithms are the most successful kinds of evolutionary algorithms which were investigated by John Holland in 1975 [16]. In Swarm Intelligence methods, instead of competition and selection, the algorithm tries to improve the position of each possible solution (or individual) through interactions among all of the solutions. In fact, in swarm intelligence methods, the solutions cooperate with each other to search the different areas of the solution space and find the best possible solution [17]. The term "Iteration" is used in swarm intelligence methods while "generation" is commonly used in evolutionary algorithms. In Fig. 2, the general procedure of swarm intelligence methods is shown. Several kinds of swarm intelligence algorithms have been proposed up to now such as particle swarm optimization (PSO) [18], inclined planes system optimization (IPO) [19], gravitational search algorithm (GSA) [20], ant colony optimization (ACO) [21], and etc. In big datasets, including lots of samples with lots of features, detecting the number of clusters and also assigning the samples to the correct clusters, are hard tasks. So, traditional clustering methods, such as K-means, are not suitable options for big data clustering. In this paper, a novel automatic clustering method based on particle swarm optimization algorithm is introduced which has shown a great performance in finding the number of clusters and also the position of the centroids in several experiments. The performance of the method is tested on several artificial datasets with different characteristics. The achieved results, especially for datasets including lots of features, demonstrate the superiority of the proposed method over traditional clustering methods. Also, the proposed method has given interesting results on a biological and 2 big mobility datasets containing almost 79000 car trajectories. The paper is organized as the following manner: in Section 2 the related works are reviewed. After that in Section 3, particle swarm optimization algorithm is described thoroughly. Then, the proposed method is completely explained in Section 4. Sections 5 and 6 are devoted to results on synthetic and biological datasets, respectively. In Section 7 the results for the real mobility datasets is interpreted. Finally, Section 8 presents the results and discussion.

Related Works

Cui et al. has introduced a hybrid method for clustering text documents which is a combination of PSO and K-means in 2005 [22]. In the first stage, PSO performs a globalized searching and in the next step K-means performs a local search. The output of the PSO, in

the first step is used as the initial points for the K-means in the second stage. In this method, average distance of data points from their corresponding centroids is used as fitness function and number of clusters is given to the algorithm as input. In 2006, Omran et al. proposed a new segmentation method based on particle swarm optimization algorithm, namely, DCPSO [23]. In this methodology, first, a pool of centroids from the data points is selected and after that the best set of centroids among them will be found. They used and analyzed different validity indices for evaluating the particles such as Dunn index and compared the performance of PSO with GA in this research. Abraham et al. introduced a clustering algorithm called MEPSO in 2008 [24]. They used Xie-Benni index for evaluating the quality of clusters. The algorithm was tested on 4 synthetics and 2 real-world datasets. Also, they used it for image segmentation. Ahmadyfard and Modares, in 2008, proposed a clustering method by combining PSO and K-means called PSO-KM [25]. First, they run PSO algorithm to search the solution space globally and then they run K-means to search around the global best solution. Again, finding the proper number of clusters is not considered in their research. Zhang and his colleagues proposed a clustering method in 2010 [26]. The suggested method was based on artificial bee colony (ABC) optimization algorithm [27]. They used ABC algorithm to find the best possible centroids and a total mean-square quantization error for evaluating the solutions. The proposed method is unable in finding the proper number of clusters. In 2014, Krishnasamy et al. proposed a clustering method using a new heuristic algorithm called cohort intelligence (CI) [28]. This algorithm is inspired from natural and society tendency of cohort candidates of learning from one another [29]. The proposed algorithm benefits from the advantages of both K-means and a modified version of CI (MCI). This combination allows the proposed algorithm to converge more quickly. In 2015, a clustering method, based on genetic algorithm, was introduced by Razavi and his colleagues [30]. They encoded the chromosomes so that each chromosome includes n number of genes which is equivalent to the number of data points in the dataset. Each gene holds the label of its corresponding sample. In the case of dealing with big datasets with huge number of samples, the length of the chromosomes will be very high and because of the large scale optimization problem, the algorithm may fail to find the best solution in a reasonable time. Also, high computation and complexity of GA compared to SI algorithms makes it slower for big datasets. Lu et al. proposed an improved K-means using a heuristic algorithm called Tabu search (TS) in 2018 [31]. In fact, they used Tabu search to overcome the drawbacks of K-means including the effect of random initial centers and getting stock to local optimum point. But still their method is unable to find the number of clusters. Fahad et al. proposed a clustering method based on a new heuristic algorithm called Grey Wolf Optimization (GWO) algorithm for vehicular ad-hoc networks [32]. The GWO algorithm is inspired by the social behavior of wolfs.

1- Initialize and evaluate the population.
2- Perform competitive selection.
3- Apply evolutionary operators (mutation and recombination)
4- Calculate the fitness amount for each solution.
5- Finish if the stopping criteria is satisfied otherwise go to step 2.

Fig. 1: Procedure of the evolutionary algorithms.

1- Generate random solutions according to the constraints.
2- Initialize the individuals.
3- Evaluate the fitness of individuals.
4- Main loop:
- For all individuals do
  I. Change the position of the individuals to form the population.
  II. Evaluate the fitness of the individuals.
  III. Find solutions with better fitness values.
  IV. Update the best solution.

Fig. 2: General procedure of swarm intelligence algorithms.

Table 1: Brief comparison between the related works

Method	Application	Automatic Clustering	Year
HYBRID PSO+K-Means [22]	Document clustering	NO	2005
DCPSO [23]	Image segmentation	NO	2006
MEPSO [24]	Image segmentation	NO	2008
PSO-KM [25]	Clustering	NO	2008
ABC Clustering [26]	Clustering	NO	2010
CI-Clustering [28]	Clustering	NO	2014
GGA [30]	Big data clustering	YES	2015
TS-KM [31]	Clustering	NO	2018
GWO-Clustering [32]	Vehicular ad-hoc network	NO	2018

They compared their method with the other heuristic algorithms. The main drawback of most methods mentioned above is that they are not designed for finding the number of clusters which is very important in big data clustering. The main novelty of this research is proposing a new accurate method based on PSO for automatic clustering. In Table 1, a brief comparison between the mentioned researches is shown.

Particle Swarm Optimization Algorithm

Particle swarm optimization algorithm, searches the solution space using a population of individuals. The algorithm is inspired by the meaningful movement of different species of the animals like birds' flocks searching for corn. Unlike the evolutionary algorithms, the population members, which are called particles, survive until the end of the process. They cooperate with each other to search the solution space and find the optimum point of the objective function. At each iteration each particle, represents a possible solution for the objective function. The position of each particle is changed by the interaction with the other particles and after that evaluated using the fitness function. In addition to the position, each particle contains a vector (*v*) which shows its velocity. Also each particle has a memory for preserving its best position from the beginning of the process to the current iteration. In each iteration, the best solution (best particle) which has the best fitness amount is considered as the leader of the population. Equations 1 and 2 show how particles move.

$$v_{id}^{t+1} = w\, v_{id}^{t} + C_1.rand.(p_{best}^{d} - x_{id}^{t}) + C_2.rand.(p_{gbest}^{d} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t} \quad (2)$$

where, v_{id} is the *d*th dimension of the velocity of the *i*th particle, *x* denotes the position of the particle, *t* is the number of iteration, *rand* is a positive random number between 0 and 1 under normal distribution, *w* is the inertia weight coefficient, p_{best} is the best position of the particle from the beginning to current iteration and p_{gbest} shows the position of the leader in each iteration. So, their movement is affected by two factors: p_{best} and p_{gbest}. The general procedure of the algorithm is shown in Fig. 3. C_1 and C_2 are two controlling factors which are called social and cognitive factors, respectively. They define whether the particle moves through p_{best} or p_{gbest}. These two factors control the exploring and exploiting ability of the algorithm in searching the solution space.

In fact, C_1 and C_2 are important in searching the solution space efficiently. According to (1) and (2), particles move toward the position of the best particle in the population (p_{gbest}) if C_2 is high and C_1 is low. However, if C_1 is high and C_2 is low, particles search around their best positions observed from the beginning to the current iteration (p_{best}).

1- Generate random position for each particle.
2- Evaluate each particle.
3- Finding the leader.
Main loop:
  A) Calculate velocities.
  B) Update position of each particle.
  C) Evaluate particles.
  D) Update P_{best} and P_{gbest}.

Fig. 3: procedure of particle swarm optimization algorithm.

Proposed Method

A. PSO-Clustering algorithm

As mentioned before, many real world problems can be formulized as optimization problems. Clustering is the process of dividing the data points in a dataset into different clusters based on their similarities so that the objects inside a cluster have the most similarity with each other and the most dissimilarity with other clusters' objects. Based on this, a new methodology for clustering can be designed using PSO. For this purpose, the particles should be encoded suitably and also a suitable fitness function for evaluating the particles should be used. Searching to find the best centroids of the dataset, each particle should contain the position of a specified number of clusters. In other words, each particle is a solution to the clustering problem which divides the dataset into different partitions by assigning data points to the closest centroid of the particle. So, the length of each particle is $k \times p$ where k is the number of clusters and p is the number of features in the dataset. Fig. 4 shows a particle with n centroids for a 2-D dataset. In this Figure, C_{ij} is the j th dimension of the i th centroid.

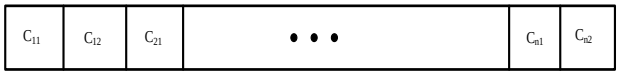


Fig. 4: A particle containing n centroids for a 2-D dataset.

After partitioning the data points in to k groups based on their distances to the centroids, it is necessary to measure the quality of the partitioning. Several indexes are introduced which can be used for this purpose like Silhouette index [33], Davies-Bouldin index [34], Dunn index [35], Calinski-Harabasz index [36] and etc. These indexes can be used as fitness function in the PSO algorithm. In this research, Calinski-Harabasz index is chosen due to its low complexity in compare to Silhouette and also its effectiveness in finding the

number of clusters. This index calculates the quality of clustering by the following equations:

$$VRC = \frac{SS_B}{SS_W} \times \frac{N-K}{K-1} \quad (3)$$

$$SS_B = \sum_{i=1}^{k} n_i . (m_i - m)^2 \quad (4)$$

$$SS_w = \sum_{i=1}^{k} \sum_{x \in c_i} (x - m_i)^2 \quad (5)$$

In these equations, SS_B is the overall between-cluster variance, SS_w is the overall within-cluster variance, *k* is the number of clusters, *N* is the number of data points, m_i is the centroid of the *i*th cluster, m is the overall mean of the sample data, x is a data point, c_i is the *i*th cluster and $(m_i - m)$ is the Euclidean distance between two vectors. Better clustering quality gives higher amount of VRC. In fact, well defined-clusters have a large SS_B and a small SS_w. So, finding the best solution for the clustering problem, PSO should search the solution space for a solution with the highest amount of VRC. Hence if we choose our fitness function as $f = \frac{1}{VRC}$, we can find the best possible solution of the clustering problem. In other words, minimizing f, PSO can find the solution which gives the highest amount of VRC, which is the optimum point.

PSO starts to search the solution space using a random population. In this problem, each population member (particle) contains the position of a predetermined number of clusters (*k*). Generating the positions randomly without considering the data points, may result in producing centroids far from the data points. This can reduce the accuracy of the final result. Actually, far centroids make the PSO algorithm search some parts of the solution space which are really far from the optimum point and prevents it from searching the closer areas to the optimum point, more effectively. In this situation the final result may not contain perfect centroids. To solve this problem, for each particle *k* samples from the dataset are randomly selected and their positions are considered as the centroids of the particle. In the next step, the quality of each particle is evaluated. For this purpose, first, all of the data points in the dataset are assigned to their closest centroid, then the quality of the particle (or its fitness value) is calculated using *Calinski-Harabasz* index. After calculating the fitness value for each particle and finding the leader, the particles move in the space through (1) and (2). Then, for each particle, after movement, a set of samples is selected randomly from the dataset, and each centroid of the particle is replaced with the position of the closest sample of this subset. In fact, a new step is added to the standard PSO which makes it search the solution space for clustering problem, more effectively.

Fig. 5, from a to d, is an example which shows how the position of a particle changes in the solution space for a 2-D dataset with 3 clusters. The procedure of the method is shown in Fig. 6. By controlling C_1 and C_2, we can search the solution space more effectively and also we can escape from the local optimum point. According to (1), a high amount of C_1 increases the effect of p_{best} and a high amount of C_2 increases the effect of p_{gbest} in searching the solution space. To search different areas of the solution space and to prevent from premature convergence, the amounts of C_1 and C_2 are changed dynamically during the search process. So that at the beginning iterations C_1 has a high amount while C_2 has a low amount and their amount change exponentially during the process. Using this strategy, in the beginning iterations the particles are scattered in the solution space in order to explore different parts of the solution space and in the last iterations they will exploit the area close to the best solution found from the beginning of the search process, to find better solution.

B. Automatic PSO-Clustering algorithm

Finding the appropriate number of clusters is very important especially in big data clustering. In fact, one of the drawbacks of conventional clustering methods, such as K-means, is their limitation in finding the number of clusters.

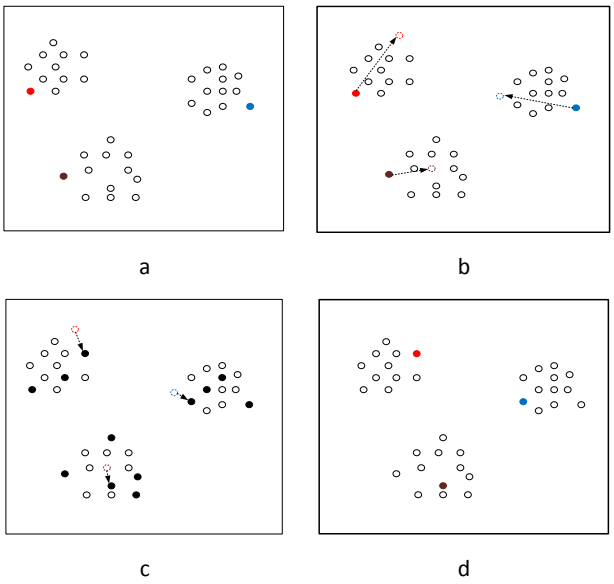


Fig. 5: From a to d: movement of the particles in the solution space for a 2-D dataset with 3 clusters.

However, a method, called X-means [37] has been introduced to cover this drawback, but its accuracy is very low which is also observed in our experiments.

Therefore, designing an accurate automatic clustering method is really necessary.

Inputs: number of solutions, maxit
Output: best solution
Initializing:
For i= 1 to number of solutions do:
select k data points randomly from the data set as centroids.
End.
Calculating the fitness amount:
For i= 1 to number of solutions do:
1. Assign each data point in the dataset to the closet centroids.
2. Calculate Calinski-Harabasz index.
End.
find the best solution.
Main loop:
For it=1 to maxit do:
For i= 1 to number of solutions do:
1. Update the position of the solutions.
2. Select j samples from the dataset randomly.
3. Find the closet samples to the new position of the centroids.
4. Assign the position of the closet samples of the position of the centroids.
5. Calculate fitness amount of the solutions.
End.
Update best solution.
Update C_1 and C_2
End.

Fig. 6: The procedure of PSO-clustering.

For this purpose, another level of searching for detecting the number of clusters is added to the PSO-Clustering methodology, introduced in the previous section. So, the automatic proposed method has two stages. In the first stage, the proposed method tries to find the number of clusters while in the second stage the main goal is finding the exact position of the centroids. At the beginning of the sequence, a random integer number is generated between 2 and $\sqrt{n}$ as k (number of clusters). Then, PSO-clustering method is run to find k centroids. After that, again PSO-clustering is run to find k_{new} centroids where k_{new} is the output of the following equation.

$$k_{new} = k_{old} \pm \varepsilon \tag{6}$$

In this equation ε is a random integer number. After running the PSO-clustering in the second step with k_{new} and the generated population, the best result including k and the fitness value of the best particle found by the algorithm, is saved. This procedure continues until the end of the sequence. In fact, in each step of the sequence the quality (fitness value) of the best particle, found by PSO-clustering method, is compared with the best result found in the previous steps. If it is better, then k is updated using (6). In our experiments, 3 sequences, each one contains 10 steps, have been used to find the number of clusters. In the second stage, again PSO-clustering method is used to search the solution space, with high number of iterations, to find the exact positions of k centroids while k is the output of the first stage. Generally, PSO-clustering is the basic building block of the proposed method used in both stages. The procedure of the APSO-clustering (Automatic PSO-clustering) is shown in Fig. 7.

Simulations and Experimental Results on Synthetic Datasets

To evaluate the performance of the proposed method, it is tested on 13 synthetic datasets [38] with different characteristics indicated in Table 2. The achieved results are reported in subsections *C* and *D* for automatic and non-automatic clustering. The accuracy of the method is calculated using rand and normalized mutual information (NMI) indexes [39]- [40] and compared with the accuracy of other clustering methods such as K-means and X-means.

Table 2: Characteristics of the datasets

	Dataset	Number of data points	Number of features	Number of clusters
Datasets S	S1 S2 S3 S4	5000	2	15
Datasets A	A1	3000	2	20
	A2	5250	2	35
	A3	7500	2	50
Datasets G2	G2-32-60	2048	32	2
	G2-128-60	2048	128	2
	G2-256-60	2048	256	2
	G2-1024-70	2048	1024	2
High dimensional datasets	Dim032	1024	32	16
	Dim064	1024	64	16

A. Rand index

Rand index measures the similarity between the partitioning achieved by a clustering algorithm and the real partitioning by comparing each pairs of data points in the dataset. Rand index is calculated using the following equation:

$$R = \frac{a+b}{a+b+c+d} \tag{7}$$

where, a is the number of pairs of data points which are in the same clusters and the algorithm also put them in the same cluster, b is the number of pairs of data points which are in different clusters and also the

algorithm put them in different clusters, c is the number of pairs of data points which are in the same clusters but the algorithm put them in different clusters wrongly, and d is the number of pairs of data points which are in different clusters but the algorithm put them in the same clusters wrongly.

Stage 1:
Inputs: number of trees, length of trees, number of population.
For i=1 to number of trees do:
1. k=Generate a random integer number
2. Beginning population=Generate a random population
3. Current state.fitness=inf
4. Current state.k=0
For j=1 to length of tree do:
If (j==1) do:
Current state.k=k
End
Best fitness=PSOclustering (k, beginning population)
If best fitness < current state.fitness
Current state.fitness=best fitness
Current state.k=k
End
K=current state.k±ε
End
Bestk[i] = current state.k
End
Output: find the best solution which has the best fitness amount and its corresponding k and its corresponding beginning population
Stage 2:
Final output= PSOclustering (best k, beginning population)

Fig. 7: The procedure of APSO-Clustering.

B. NMI index

NMI index calculates the mutual information between two variables. Actually, these two variables are the labels of the data find by a clustering algorithm and the real labels. The concept of mutual information is linked to that of entropy. It means that this index calculates an entropy function to find the amount of mutual information between two variables. The NMI index is calculated using the following equation:

$$NMI(Y,C)=\frac{2\times I(Y;C)}{H(Y)+H(C)} \tag{8}$$

where Y is the real labels of the data, C is the output of the clustering algorithm, H is the entropy function and I is the function defining the mutual information between Y and C. The following equations show how H and I are calculated:

$$H=-\sum_{i=1}^{n} p_i \log p_i \tag{9}$$

C. Results for non-automatic clustering

Tables 3 to 6 contain the average accuracy of different methods on S, A, G2 and high dimensional datasets respectively. In these tables the accuracy is calculated using *Rand index.* Also Figs. 8, 9, 10 and 11 show centroids of the best solutions found by our algorithm for these artificial datasets. In Fig. 11 the data points are plotted based on their first and second features. In these figures the gray stars are the centroids of the best solutions (particles) found by the algorithm. According to the tables and figures, PSO-clustering method not only overcomes K-means in the most of the experiments, but also it finds the best possible position of the centroids. According to Table 5 and Table 6, K-means has shown a good performance on G2 and high-dimensional datasets, respectively. One reason is that K-means does not search the solution space to find the number of clusters. It only tries to detect the position of the centroids. Generally, K-means has an easier task in compare to the proposed method. On the other hand, it should be mentioned that the artificial datasets, used in this research, are not big datasets. These datasets are selected to evaluate the performance of the proposed method when dealing with different aspects of big datasets. For example, datasets S, have moderate number of clusters but the clusters are overlapped which makes it hard for the clustering algorithms to distinguish the clusters. Datasets A have high number of clusters while G2 and high dimensional datasets contain high number of features. According to the tables although K-means has shown a good performance but the proposed method has overcome K-means in all these situations.

Fig. 12 contains the convergence curves of PSO-clustering method achieved for two datasets. In these figures, the blue curves show the fitness amount of the best particle in each iteration and the red curves demonstrate the average fitness amount of all particles in each iteration. Far distance between these curves means that the particles are located in different parts of the solution space while low distance means that they are close to each other and located around the best particle. In other words, these figures indicate that in the first iterations, the algorithm searches different parts of the solution space and in the last iterations it searches the area close to the best solution found from the beginning of the process. This means that PSO-clustering method searches the solution space effectively and controlling the exploring and exploiting parameters can prevent premature convergence and being trapped in local optimum point.

In Fig. 13, the ROC graph for S1 dataset is shown. This curve is plotted based on [41]. In fact, this curve is plotted by calculating the pair of [type1 error, type2 error] for different number of K. It indicates that

increasing the number of clusters causes a reduction in type1 error.

D. Results for automatic clustering

The main challenge in clustering, especially in big data clustering, is finding the number of clusters. In most of the conventional clustering algorithms, such as K-means, the number of clusters should be given to the algorithm as an input while in the most cases there is no information about the dataset and it is impossible to distinguish the number of clusters by visualization.

So, designing an accurate clustering algorithm which not only has high ability in finding the position of the centroids but also it is very accurate in finding the number of clusters, is really needed. In the next pages, the result of our APSO-clustering algorithm, able in finding the number of clusters, is shown and compared with X-means, which is also an automatic clustering method.

Tables 7 to 14 show the average results achieved by APSO and X-means for S, A, G2 and high dimensional datasets, respectively. Table 7 and Table 8 show the performance of our method and X-means on datasets S, respectively. Each of these datasets (S1, S2, S3, S4) have 15 Gaussian clusters with different overlapping ranges [42].

From S1 to S4, the range of overlapping of different clusters is increased. This means that S4 has 15 overlapped clusters while S1 has 15 distinct clusters. This is clearly demonstrated in Fig. 8. According to Table 7, the reduction of Rand index, from S1 to S4, is not remarkable while the values of NMI index show a remarkable reduction from S1 to S4. So, for these datasets, NMI index shows the difference between APSO-clustering and X-means more clearly. According to these two tables, APSO-clustering has overcome X-means in all four datasets.

Table 3: Accuracy of three methods for datasets S

Method	S1	S2	S3
PSO-clustering	0.998	0.992	0.965
Standard PSO	0.988	0.985	0.955
K-means	0.990	0.977	0.9522

Table 4: Accuracy of three methods for datasets A

Method	A1	A2	A3
PSO-clustering	0.9993	0.9994	0.9966
Standard PSO	0.9617	0.9811	0.9889
K-means	0.9877	0.9924	0.9949

Table 5: Accuracy of three methods for datasets G2

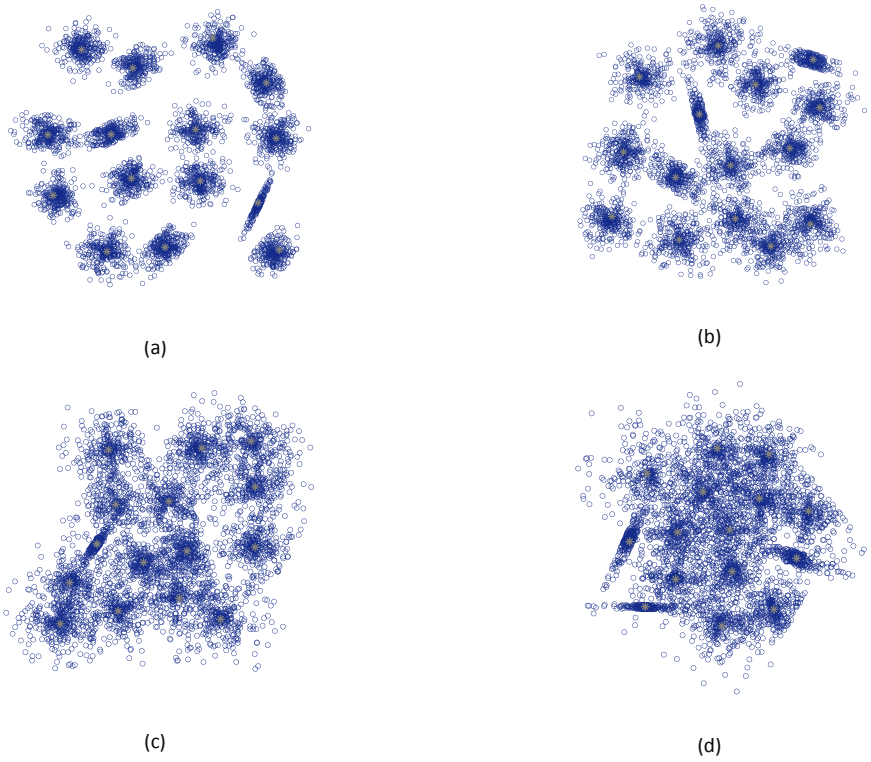
Method	G2-32-60	G2-128-60	G2-256-60	G2-1024-70
PSO-clustering	1	1	1	1
Standard PSO	0.851	0.9422	0.8998	0.9095
K-means	1	1	1	1

Table 6: Accuracy of three methods for high dimensional datasets

Method	Dim032	Dim064
PSO-clustering	1	1
Standard PSO	0.9576	0.966
K-means	1	1

Table 9 and Table 10 show the performance of the two automatic methods on A datasets. The main property of these datasets is that they contain high number of Clusters [43]. For these datasets, APSO-clustering has shown a great accuracy in finding the number of clusters while the accuracy of X-means is weak. The values of Rand and NMI indexes confirm the superiority of APSO-clustering over X-means. Table 11 and Table 12 indicate the great performance of the proposed method on datasets G2. All these datasets have 2 Gaussian clusters with different standard deviation ranges [44]. For example, G2-1024-70 has 2048 samples with 1024 features in 2 clusters and the standard deviation of the samples for this dataset is 70 while for the other 3 datasets the standard deviation is 60. As shown in Table 11, for G2-1024-70 dataset, which has 1024 features, APSO-clustering method has found the exact number of clusters while X-means has found 17 clusters. For this dataset, each particle at least has 2048 cells.

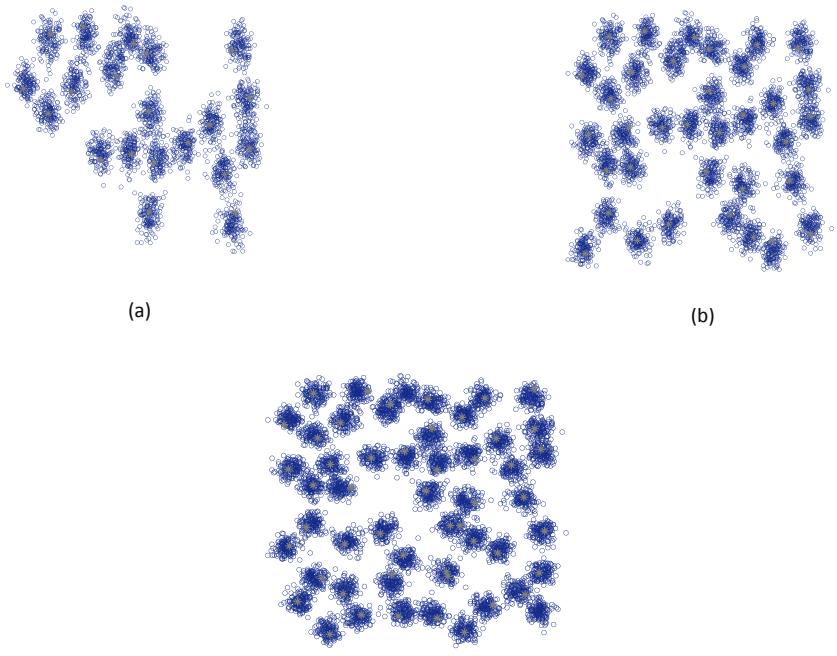
This is a high dimensional optimization problem which only can be solved by heuristic or meta-heuristic algorithms. According to Table 11 and Table 12, Rand and NMI indexes are 100% for APSO-clustering and 50% and 30% for X-means, respectively. These numbers indicate that the proposed method not only finds the exact number of clusters, but also it finds the positions of the centroids accurately. Table 13 and Table 14 have the performance of the methods on high dimensional datasets. These datasets have 1024 samples in 16 distinct clusters [45]. According to these tables, although X-means has shown a good performance but still Rand and NMI indexes for X-means are lower than APSO-clustering method. Also for these datasets the proposed method has shown a great accuracy in finding the number of clusters.



(a) (b)

(c) (d)

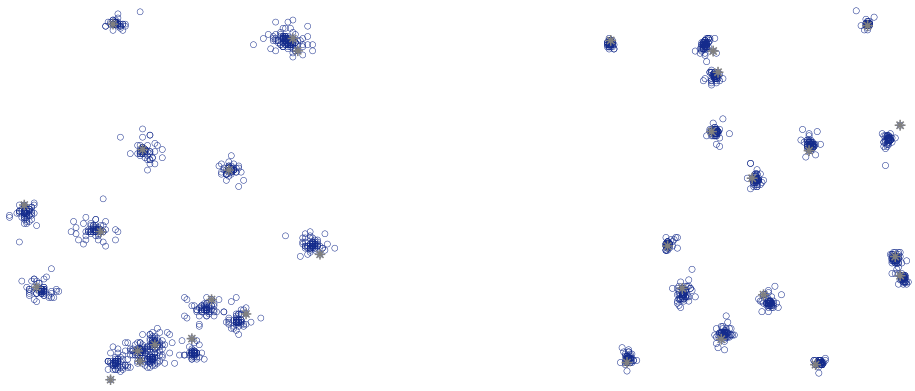
Fig. 8: a, b, c, d: Results for S1, S2, S3 and S4, respectively.



(a) (b)

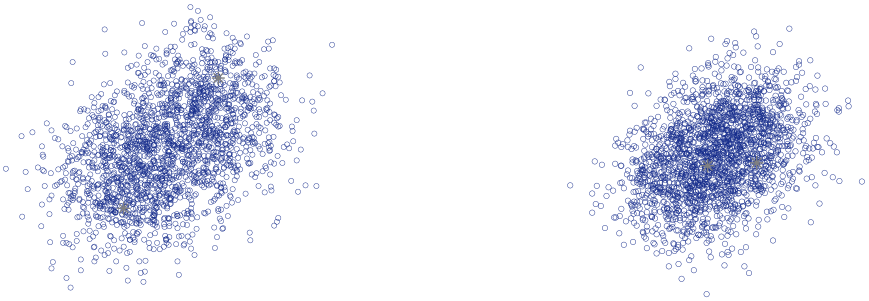
(c)

Fig. 9: a, b, c: Results for A1, A2 and A3, respectively.



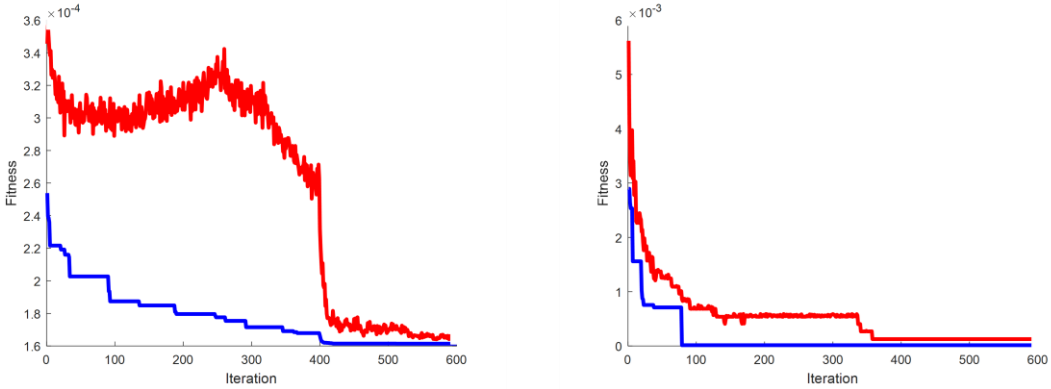
(a) (b)

Fig. 10: a, b: Results for dim032 and dim064 datasets, respectively.



(a) (b)

Fig. 11: a and b: Results for G2-256-60 and G2-1024-70 datasets, respectively.



(a) (b)

Fig. 12: a and b: Convergence curves of PSO-Clustering method for S4, and dim064 datasets, respectively.

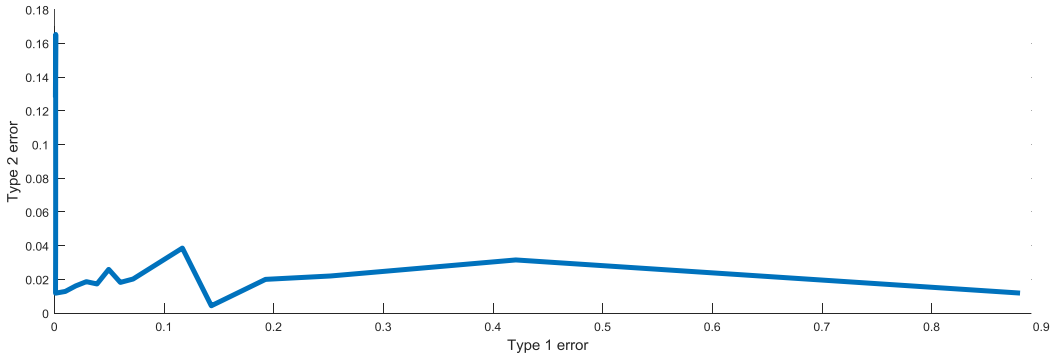


Fig. 13: ROC curve for S1 dataset.

Table 7: Average results achieved by APSO-CLUSTERING for datasets S

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
S1	0.9975	0.9831	15	15.5	614.375
S2	0.9898	0.9345	15	16.5	609.695
S3	0.9658	0.7952	15	15	542.27
S4	09545	0.7196	15	15.5	628.804

Table 8: Average results achieved by X-means for datasets S

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
S1	0.9161	0.8094	15	8	67046.73
S2	0.9361	0.8110	15	9	64508.38
S3	0.9156	0.718	15	9	63312.01
S4	0.9199	0.666	15	10	64723.92

Table 9: Average results achieved by APSO-CLUSTERING for datasets A

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
A1	0.9969	0.9778	20	21.5	417.255
A2	0.9981	0.98	35	36.5	683.72
A3	0.9966	0.9717	50	47	908.77

Table 10: Average results achieved by X-means for datasets A

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
A1	0.8569	0.728	20	6	19195.98
A2	0.908	0.743	35	9	76949.35
A3	0.901	0.715	50	9	183845.54

Table 11: Average results achieved by APSO-CLUSTERING for datasets G2

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
G2-32-60	1	1	2	2	495.404
G2-128-60	1	1	2	2	665.686
G2-256-60	1	1	2	2	996.472
G2-1024-70	1	1	2	2	3299.48

Table 12: Average results achieved by X-means for datasets G2

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
G2-32-60	0.5	0.313	2	10	10139.59
G2-128-60	0.5	0.309	2	11	16680.01
G2-256-60	0.5	0.308	2	15	22715.12
G2-1024-70	0.5	0.307	2	17	70644.21

Table 13：Average results achieved by APSO-CLUSTERING for high dimensional datasets

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
Dim032	0.9998	0.9984	16	17	493.258
Dim064	0.9999	0.9997	16	16.33	511.483

Table 14: Average results achieved by X-means for high dimensional datasets

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
Dim032	0.984	0.968	16	14	1899.7
Dim064	0.984	0.968	16	14	2147.8

Table 15: Average results achieved by parallel APSO-CLUSTERING for datasets S

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
S1	0.9966	0.9797	15	16	266.418
S2	0.9992	0.9464	15	15	258.9
S3	0.9613	0.7833	15	14.66	246.708
S4	0.9548	0.797	15	16	303.845

Table 16: Average results achieved by parallel APSO-CLUSTERING for datasets a

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
A1	0.9952	0.9758	20	20	233.845
A2	0.9975	0.9794	35	36	330.571
A3	0.9963	0.9642	50	57.66	527.877

Table 17: Average results achieved by parallel APSO-CLUSTERING for datasets G2

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
G2-32-60	1	1	2	2	185.853
G2-128-60	1	1	2	2	320.101
G2-256-60	1	1	2	2	534.737
G2-1024-70	1	1	2	2	2009.9

Table 18: Average results achieved by parallel APSO-CLUSTERING for high dimensional datasets

Dataset	Rand index	NMI index	Real number of k	Detected number of k	Run time (second)
Dim032	0.9999	0.9992	16	16.5	231.903
Dim064	0.9989	0.9936	16	19.5	254.234

Generally, these tables indicate the superiority of our APSO-clustering method over X-means both in finding the number of clusters and in finding the positions of the centroids. Also, based on run times, written in these tables, our method is much faster than X-means which is really important in big data clustering. Fig. 14 a and b show the centroids of the best particles found for S2 and A3 datasets respectively where the algorithm didn't find the exact number of clusters. In these figures the red stars are the extra centroids detected by the algorithm. According to Table 7, Table 9 and Fig. 14, although the algorithm did not find the exact number of clusters, the accuracy of the proposed method is very high. In other words, the algorithm's performance is remarkable both in finding the number of clusters and the position of the centroids. Fig. 15, shows the convergence graphs for two datasets. These graphs indicate that the proposed method searches the solution space thoroughly. The blue curves which show the position of the best particle, have been improved during the search process. This phenomenon demonstrates the power of APSO-clustering method in escaping from local optimum points.

E. Results for parallel APSO-clustering method

In addition to accuracy, the other important factor that should be considered especially for big data clustering, is the run time. Achieving perfect results not in a reasonable period of time is unacceptable. In big data analytics, one of the important things is the possibility of parallelizing the analytical algorithm. Fortunately, the first stage of the proposed method can be implemented in parallel form. As mentioned in the previous sections, the first stage contains 3 independent sequences. Since they are independent, they can be run at the same time using at least 3 processing units. This can reduce the run time of the proposed method. In this section, the results of the parallel APSO-clustering algorithm on synthetic datasets, are presented.

The parallel algorithm is implemented using MATLAB parallel computing toolbox on a single machine with 3 local cores. The following line shows the properties of the machine:

CPU: Core i7-4700 MQ, 2.4 GHz, RAM: 8GB

Tables 15 to 18 contain the results achieved by parallel APSO-Clustering method on S, A, G2 and high dimensional datasets respectively. According to these tables, parallelizing the proposed method has caused almost 50% reduction in run time for each group of datasets, while the accuracy is remained unchanged. This is a great achievement with a single ordinary machine.

Definitely having more and stronger processing units (like HPC systems) will result in more run time reduction. In fact, these tables indicate the scalability of the proposed method for big data clustering. For example, according to Table 17 run time for G2-1024-70 dataset has a 40% reduction.

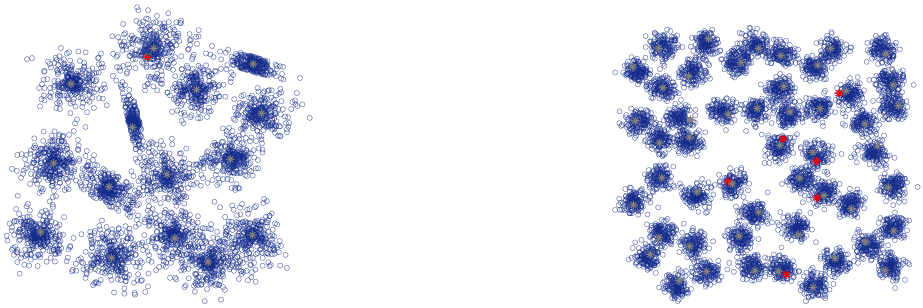
This is very impressive for a dataset with 1024 features since it is achieved by a single ordinary machine.

Performance Evaluation on a Biological Dataset

In order to complete our investigations on the performance of the proposed method on the big datasets, we have tested APSO-clustering method on a biological dataset called GSE 5847. This dataset contains experimental data from a gene expression study of tumor stroma and epithelium cells from 15 inflammatory breast cancer (IBC) cases and 35 non-inflammatory breast cancer cases [46].

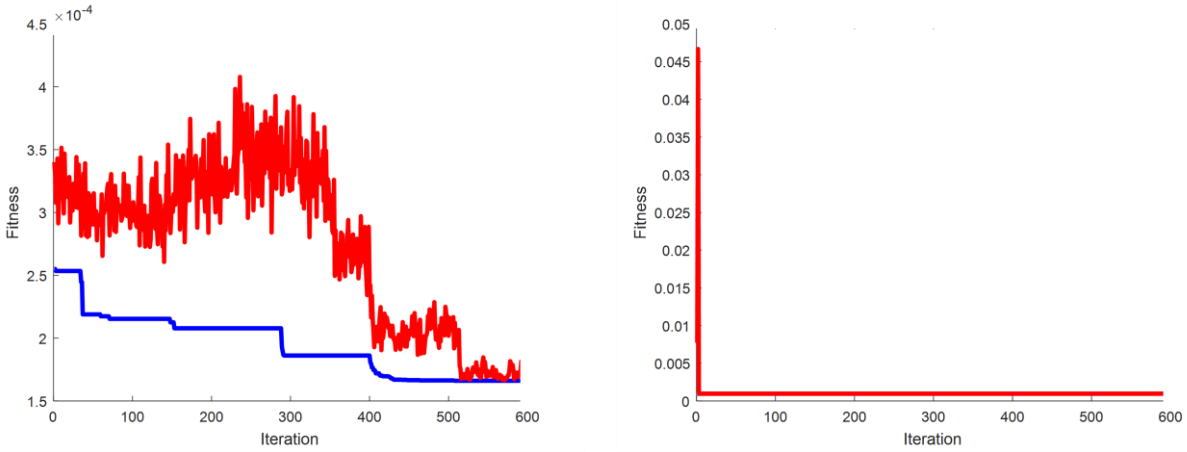
Generally, this dataset contains 22316 samples and 95 features. Table 19 shows the performance of the proposed method on this real big dataset. Since the dataset does not have label, it is impossible to calculate Rand and NMI indexes.

Therefore, Silhouette and Davies-Bouldin indexes are used for measuring the accuracy of APSO-clustering.



(a) (b)

Fig. 14: a and b: Results achieved by APSO-clustering for S2 and A3, respectively.



(a) (b)

Fig. 15: a and b: Convergence curves of APSO-clustering method for S4 and G2-1024-70 datasets, respectively.

According to this table, APSO-clustering method has found four clusters in 5016.65 seconds.

The Silhouette and Davies-Bouldin indexes amounts are 0.6037 and 0.7736, respectively.

In Fig. 16 the samples (blue circles) and the centroids (gray stars) are plotted based on the first two features.

Table 19: Performance evaluation of the proposed method on GSE5847

Dataset	Number of clusters	Silhouette index	Davies-Bouldin index	Run time
GSE5847	4	0.6037	0.7736	5016.65

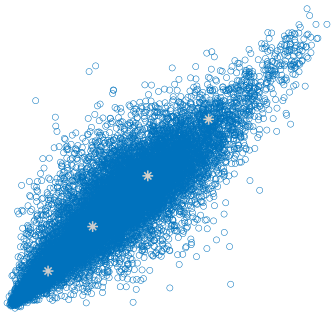


Fig. 16: Centroids found for GSM5847 dataset.

Simulations and Experimental Results on Real Big Mobility Datasets

We concentrate in this paper on massive real life GPS datasets, obtained from private vehicles with on-board GPS receivers. The owners of these cars are subscribers of a pay-as-you-drive car insurance contract, under which the tracked trajectories of each vehicle are periodically sent (through the GSM network) to a central server for anti-fraud and anti-theft purposes. This dataset has been donated for research purposes by Octo Telematics Italia S.r.l, the leader for this sector in Europe. In particular, the dataset used is about ≈40,000 cars tracked during 5 weeks (from June 14th through July 18th, 2011) in Tuscany, a 100 km × 100 km square centered on the city of Pisa. The average sampling rate of the GPS receivers is 30 seconds. Globally, the dataset is composed of almost 20 Million observations, each consisting of a quadruple (*id*, *lat*, *long*, *t*), where *id* is the car identifier, (*lat*, *long*) are the spatial coordinates, and *t* the time of the observation. The car identifiers are pseudonymized, in order to achieve a basic level of anonymity. The resolution of the spatial coordinates is at 10−6 degrees, and the error of the positioning system is estimated at 10-20 m in normal conditions. All the observations of the same car *id* over the entire observation period are chained together in increasing temporal order into a global trajectory of car id. The global trajectory is then split into several sub-trajectories, corresponding to trips or travels, by using a cut-off threshold of 30 minutes: if the time interval between two subsequent observations of the car is larger than 30 minutes, the first observation is considered as start of another travel; using this reconstruction procedure we obtained almost 1,500,000 different travels. To extract more details from the dataset we split it 7 datasets one for each day of the week. In fact, to understand if the movement patterns of the cars depend on the day of the week, we analyzed the travels in each day separately. In particular, in the following analysis we will show the results on Pisa_Monday and Pisa_Sunday datasets which have 49,000 and 29,000 trips respectively. Clearly, trips consist of different number of coordinates. In other words, each dataset contains samples with different number of features. Therefore, for simplicity of measuring the distance between the data points (trajectories or trips) we considered the first and last point of each trajectory. In fact, we considered each trajectory as an array with four elements including the latitude and longitude of the first and last point of each trajectory. We used our method (APSO-Clustering) in a hierarchical form to obtain more accurate results with more details. It means that, first we group the dataset into *k* clusters using APSO-Clustering algorithm and in the next levels for each cluster we repeat this procedure to gain more details about the trajectories. The procedure continues until the quality of the clustering doesn't show any improvement at the end of each level. Actually the quality is the value of Calinski-Harabasz index measured for the clusters found at the end of each level. The achieved results are shown in Table 20. According to this table, the algorithm in the first level partitioned the whole Pisa_Monday dataset into 2 clusters. In the next level, these 2 clusters are divided into 5 sub-clusters and in the last level the sub-clusters from the second level again divided into 28 sub-clusters. In other words, the first 2 clusters detected in the first level, can be considered as macro clusters and as the procedure goes on more details about these two macro clusters are extracted. Fig. 17 shows how the results of the algorithm represent a hierarchical exploration of the data. For Pisa_Sunday dataset, the number of clusters and sub-clusters for each level are 2, 10 and 47 respectively. In the next two subsections the details of the achieved results for the two datasets are described separately.

A. Results achieved for Pisa_Monday dataset

The macro clusters, detected in the first level, are shown in Fig. 18. In this figure the lines are trajectories and the triangles define the destination of each trajectory. According to this figure, trips have different number of points but the results are achieved based on their first and last points. Fig. 18 shows that the first macro cluster contains the trips to east and the second one contains the trips to west. In Fig. 19 and 20, 4 sub-clusters, extracted from each of these two macro clusters, are demonstrated. These sub-clusters are found in the last level. As expected, the trip's destination of sub-clusters, shown in Fig. 19, is east while the destination of the trips shown in Fig. 20, is west. According to these figures, the algorithm finds the trips with the same destination and put them in the same cluster. The next subsection includes the corresponding results for Pisa_Sunday dataset.

B. Results achieved for Pisa_Sunday dataset

According to Table 20, the algorithm has extracted 47 clusters from this dataset after three levels of searching while it has extracted 28 clusters for Pisa_Monday. It seems reasonable since Sunday is weekend. Like Pisa_Monday dataset, the algorithm has found 2 macro clusters after the first level and has divided them into 47 sub-clusters at the end of the third level. Fig. 21 contains the first two macro clusters for Pisa_Sunday dataset. In this figure, generally the first macro cluster contains the trips to west while the second one contains the trips to east. Fig. 22 and Fig. 23 include 4 sub-clusters extracted from each of the macro clusters.

Table 20: Results achieved by hierarchical APSO-CLUSTERING for Pisa_Monday and Pisa_Sunday datasets

Dataset	Number of clusters in level	Number of clusters in level	Number of clusters in level
Pisa_Monday	1	2	3
	2	5	28
Pisa_Sunday	2	10	47

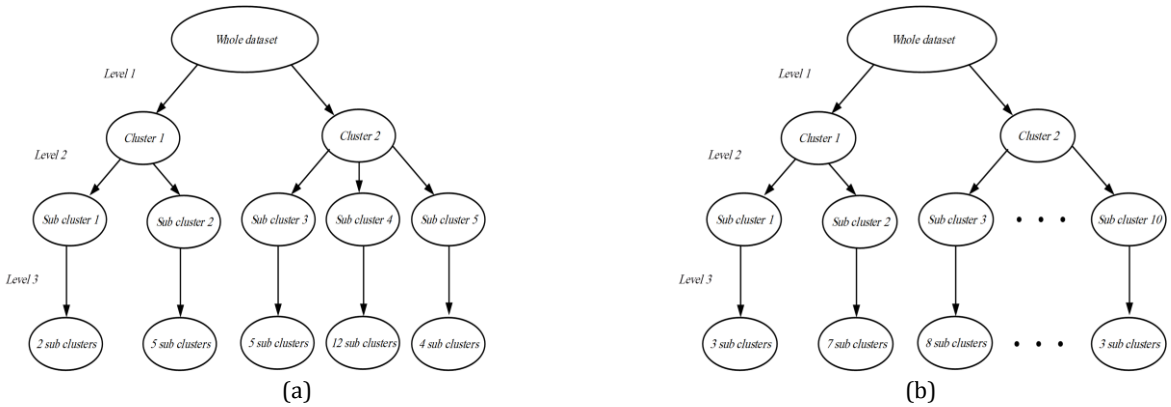


Fig. 17: a, b: The whole procedure of clustering Pisa_Monday and Pisa_Sunday, respectively.

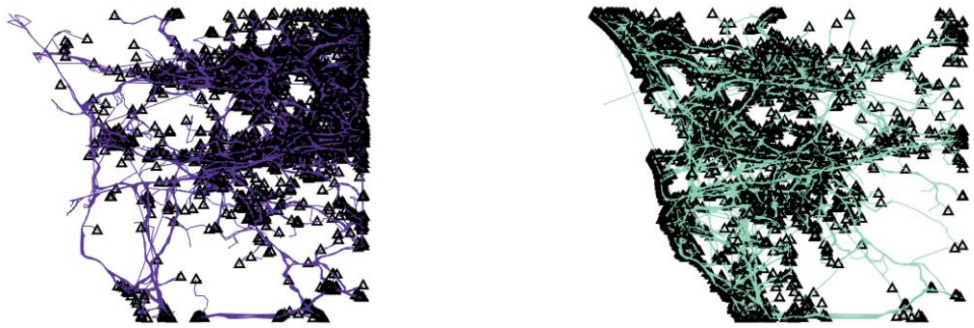


Fig. 18: a, b: The first two clusters found in the first level for Pisa_Monday dataset.

Also the sub-clusters shown in these two figures show the trips with the same destination with their corresponding macro clusters. According to Figs. 18 to 23, it can be seen that the power of the proposed method, in finding different clusters with different destinations, is remarkable which is also demonstrated in section 5.

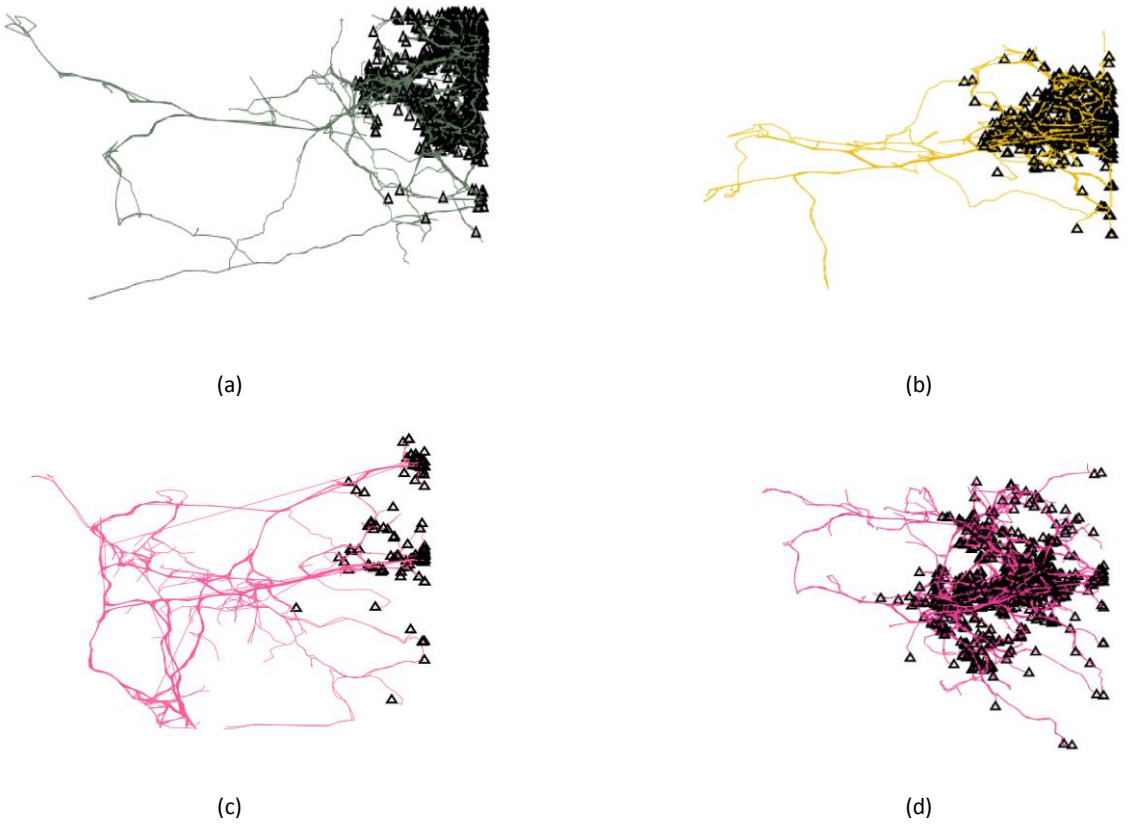
Results and Discussions

Clustering is an important data mining technique, which is the process of dividing the objects of a dataset, into different clusters.

Several techniques have been introduced for clustering up to now, like K-means.K-means is the most popular clustering algorithm which is widely used in different applications.

K-means has some drawbacks such as its tendency to converge to local optima, its dependency on the initial value of cluster centers and its inability in finding the number of clusters.

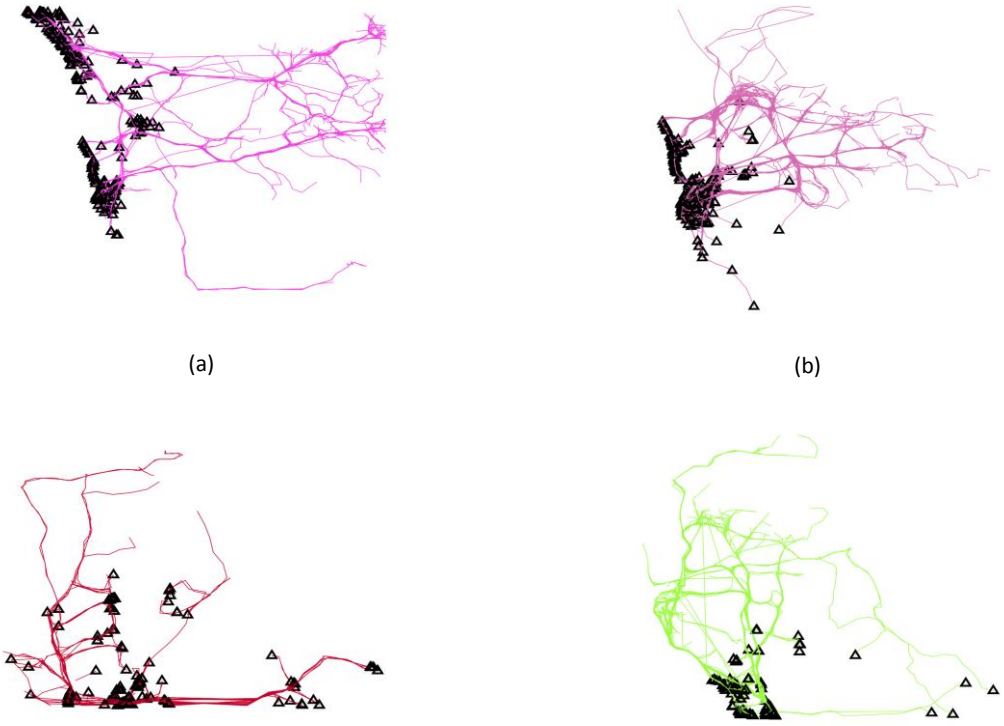
These drawbacks, prevent it from performing well for big datasets with high number of features or high number of clusters.



(a) (b)

(c) (d)

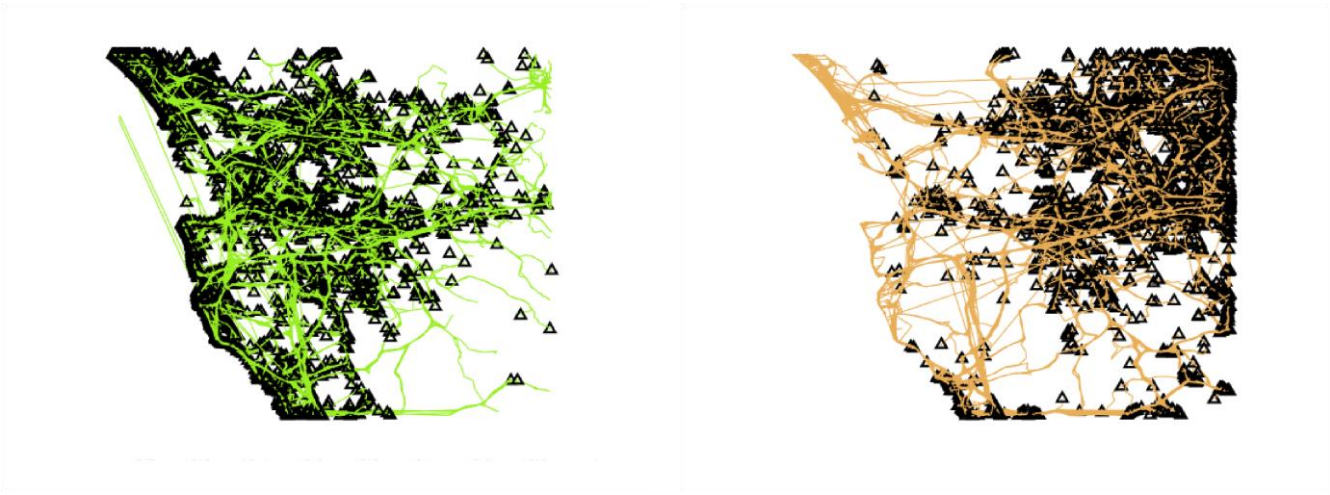
Fig. 19: Four sub-clusters extracted from the first macro cluster.



(a) (b)

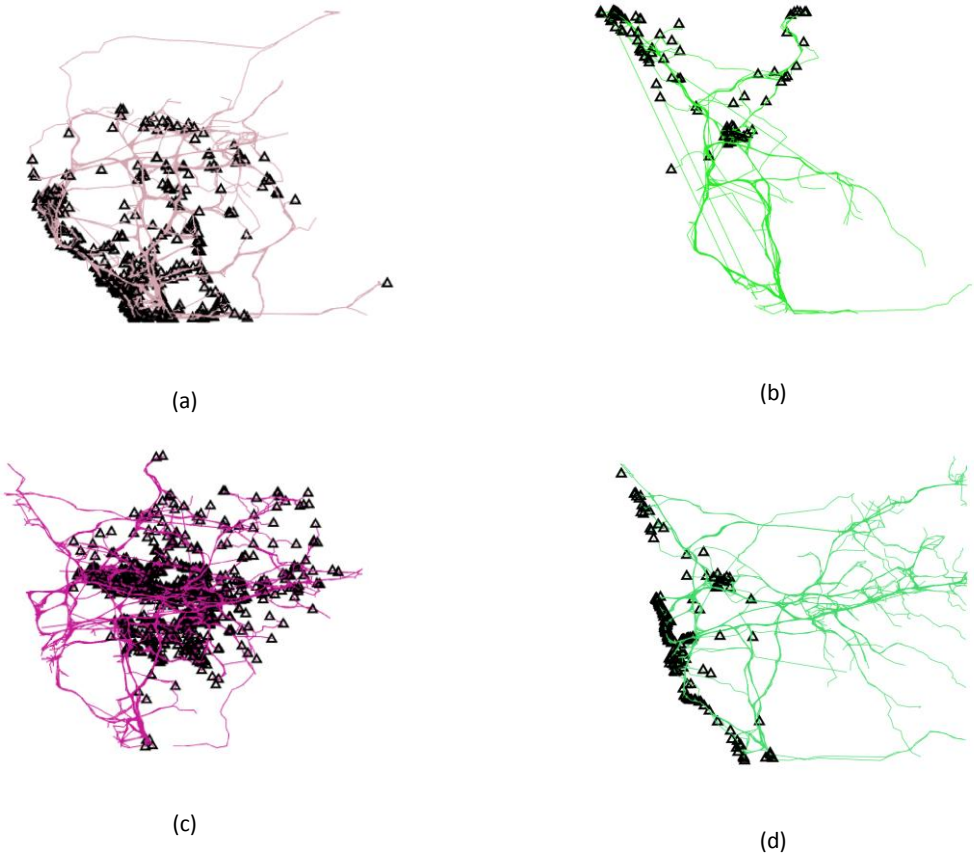
(c) (d)

Fig. 20: Four sub-clusters extracted from the second macro cluster.



(a) (b)

Fig. 21: a, b: The first two clusters found in the first level for Pisa_Sunday dataset.



(a) (b)

(c) (d)

Fig. 22: a, b, c, d: Four sub-clusters extracted from the first macro cluster.

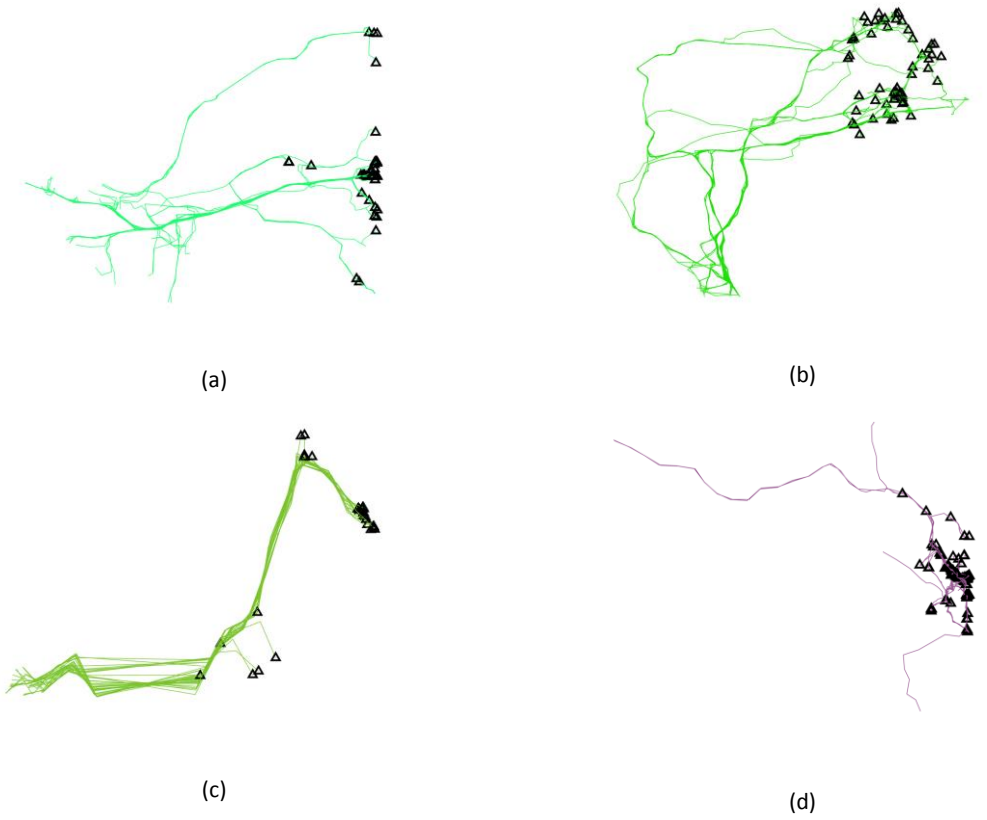


Fig. 23: Four sub-clusters extracted from the second macro cluster.

In fact, high number of objects with high number of dimensions in big datasets, make it impossible for conventional clustering techniques, such as K-means, find the correct position of centroids and the right number of clusters. However, an extended version of K-means, called X-means, has been introduced which has the ability to estimate the number of clusters, but its accuracy is not high enough especially when dealing with big datasets.

In this research, we introduced a new clustering method based on a swarm intelligence algorithm (PSO) for big data clustering.

We tested the proposed method on 13 synthetic datasets with different characteristics, a biological big dataset and 2 real big mobility datasets.

We compared its accuracy with K-means and X-means. According to the tables and figures, our APSO-clustering algorithm, not only outperforms K-means in finding the position of the centroids, but also it finds the number of clusters accurately.

Also the APSO-clustering overcomes X-means both in finding the number of clusters and the positions of the clusters. Furthermore, the proposed method is much faster than X-means. This shows its power and effectiveness in clustering.

Conclusion

The results achieved for real big mobility datasets show the power and effectiveness of the proposed method when dealing with real big datasets. In another point of view, the reported results both for synthetic datasets and real big datasets demonstrate the power and accuracy of the swarm intelligence methods in solving complex optimization problems.

Author Contributions

Iman Behravan, Dr. Roberto Trasarti and Dr. Seyed Hamid Zahiri designed the APSO-Clustering algorithm. Dr.Roberto Trasarti collected the mobility datasets and Dr.Seyed Mohammad Razavi analyzed and interpreted the results.

Acknowledgment

The authors acknowledge professor Fosca Gianotti for her help and guide through this project.

Conflict of Interest

The author declares that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancy have been completely observed by the authors.

Abbreviations

PSO	Particle swarm optimization
SI	Swarm Intelligence
EA	Evolutionary Algorithms
IPO	Inclined Planes System Optimization
GSA	Gravitational Search Algorithm
GA	Genetic Algorithm
ABC	Artificial Bee Colony
CI	Cohort Intelligence
TS	Tabu Search
GWO	Grey Wolf Otimizer
P_{gbest}	Position of the leader in PSO
P_{best}	Best position observed by the particle during the search process.
APSO-Clustering	Automatic PSO-Clustering
NMI	Normalized Mutual Information

References

[1] J. V. Aggarwal, V. Bhatnagar, D. K. Mishra, Big Data Analytics, Springer Singapore, 2018,

[2] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. Hung Byers, "Big data: The next frontier for innovation, competition, and productivity," Tech. Rep., 2011,

[3] S. Cheng, Y. Shi, Q. Qin, R. Bai, "Swarm intelligence in big data analytics," in Proc. International Conference on Intelligent Data Engineering and Automated Learning: 417-426, 2013.

[4] A. Rajaraman, J. D. Ullman, Mining of massive datasets: Cambridge University Press, 2011.

[5] S. Cheng, Q. Zhang, Q. Qin, "Big data analytics with swarm intelligence," Industrial Management & Data Systems, 116(4): 646-666, 2016.

[6] A. K. Jain, M. N. Murty, P. J. Flynn, "Data clustering: A review," ACM Computing Surveys (CSUR), 31(3): 264-323, 1999.

[7] J. A. Hartigan, "Clustering algorithms," John Wiley & Sons, 1975,

[8] R. Xu, D. Wunsch, "Survey of clustering algorithms," IEEE Transactions on Neural Networks, 16(1): 645-678, 2005.

[9] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern recognition letters, 31(8): 651-666, 2010.

[10] J. A. Hartigan, M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1): 100-108, 1979.

[11] J. C. Bezdek, R. Ehrlich, W. Full, "FCM: The fuzzy c-means clustering algorithm," Computers & Geosciences, 10(2-3): 191-203, 1984.

[12] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, T. Herawan, "Big data clustering: a review," in Proc. International Conference on Computational Science and Its Applications: 707-720, 2014.

[13] S. Cheng, B. Liu, T. Ting, Q. Qin, Y. Shi, K. Huang, "Survey on data science with population-based algorithms," Big Data Analytics, 1(1): 3, 2016.

[14] Y. Shi, "An optimization algorithm based on brainstorming process," in Emerging Research on Swarm Intelligence and Algorithm Optimization: 1-35, 2015.

[15] N. Kokash, "An introduction to heuristic algorithms," Department of Informatics and Telecommunications: 1-8, 2005,

[16] X. Yu, M. Gen, Introduction to evolutionary algorithms: Springer Science & Business Media, Springer-Verlag, London 2010.

[17] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems: 1, Oxford university press, 1999,

[18] J. Kennedy, "Particle swarm optimization," in Encyclopedia of machine learning, ed: Springer: 760-766, 2011.

[19] M. H. Mozaffari, H. Abdy, S. H. Zahiri, "IPO: An inclined planes system optimization algorithm," Computing and Informatics, 35(1): 222-240, 2016.

[20] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, "GSA: A gravitational search algorithm," Information Sciences, 179(13): 2232-2248, 2009.

[21] M. Dorigo, M. Birattari, "Ant colony optimization," in Encyclopedia of machine learning, ed: Springer: 36-39, 2011.

[22] X. Cui, T. E. Potok, P. Palathingal, "Document clustering using particle swarm optimization," in Proc. IEEE Swarm Intelligence Symposium: 185-191, 2005.

[23] M. G. Omran, A. Salman, A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," Pattern Analysis and Applications, 8(4): 332, 2006.

[24] A. Abraham, S. Das, S. Roy, "Swarm intelligence algorithms for data clustering," in Soft Computing for Knowledge Discovery and Data Mining, O. Maimon and L. Rokach, Eds., ed Boston, MA: Springer US: 279-313, 2008,

[25] A. Ahmadyfard, H. Modares, "Combining PSO and k-means to enhance data clustering," in Proc. IEEE International Symposium on Telecommunications: 688-691, 2008.

[26] C. Zhang, D. Ouyang, J. Ning, "An artificial bee colony approach for clustering," Expert Systems with Applications, 37(7): 4761-4767, 2010.

[27] D. Karaboga, B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," Applied Soft Computing, 8(1): 687-697, 2008.

[28] G. Krishnasamy, A. J. Kulkarni, R. Paramesran, "A hybrid approach for data clustering based on modified cohort intelligence and K-means," Expert Systems with Applications, 41(13): 6009-6016, 2014.

[29] A. J. Kulkarni, I. P. Durugkar, M. Kumar, "Cohort intelligence: a self-supervised learning behavior," in Proc. IEEE International Conference on Systems, Man, and Cybernetics: 1396-1400, 2013.

[30] S. H. Razavi, E. O. M. Ebadati, S. Asadi, H. Kaur, "An efficient grouping genetic algorithm for data clustering and big data analysis," in Computational Intelligence for Big Data Analysis, ed: Springer: 119-142, 2015.

[31] Y. Lu, B. Cao, C. Rego, and F. Glover, "a tabu search based clustering algorithm and its parallel implementation on spark," Applied Soft Computing, 63(63): 97-109, 2018.

[32] M. Fahad, F. Aadil, Z. u. Rehman, S. Khan, P. A. Shah, K. Muhammad, et al., "Grey wolf optimization based clustering algorithm for vehicular ad-hoc networks," Computers & Electrical Engineering, 70(1): 853-870, 2018.

[33] A. Starczewski, A. Krzyżak, "Performance evaluation of the silhouette index," in Proc. International Conference on Artificial Intelligence and Soft Computing: 49-58, 2015.

[34] D. L. Davies, D. W. Bouldin, “A cluster separation measure,” IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(1): 224-227, 1979.

[35] M. K. Pakhira, S. Bandyopadhyay, U. Maulik, "Validity index for crisp and fuzzy clusters,” Pattern Recognition, 37(3): 487-501, 2004.

[36] T. Caliński, J. Harabasz, “A dendrite method for cluster analysis,” Communications in Statistics-theory and Methods, 3(1): 1-27, 1974.

[37] D. Pelleg, A. W. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” in Proc. 17th International Conf. on Machine Learning Citations (ICML), 1(1): 727-734, 2000.

[38] School of Computing University of Eastern Finland. (2015, April 4, 2018). Clustering basic benchmark.

[39] W. M. Rand, "Objective criteria for the evaluation of clustering methods," Journal of the American Statistical association, 66(336): 846-850, 1971.

[40] A. F. McDaid, D. Greene, N. Hurley, "Normalized mutual information to evaluate overlapping community finding algorithms," arXiv preprint arXiv:1110.2515, 2011.

[41] H. Aidos, R. P. Duin, A. L. Fred, “The area under the ROC curve as a criterion for clustering evaluation," in Proc. 2nd International Conference on Pattern Recognition Applications and Methods (ICPRAM): 276-280, 2013.

[42] P. Fränti, O. Virmajoki, “Iterative shrinking method for clustering problems,” Pattern Recognition, 39(5): 761-775, 2006.

[43] I. Kärkkäinen, P. Fränti, Dynamic local search algorithm for the clustering problem: University of Joensuu, 2002.

[44] P. Fränti, R. Mariescu-Istodor, C. Zhong, “XNN graph,” in Proc. Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR): 207-217, 2016.

[45] P. Franti, O. Virmajoki, V. Hautamaki, “Fast agglomerative clustering using a k-nearest neighbor graph,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(11): 1875-1881, 2006.

[46] B. D. Lehmann, J. A. Bauer, X. Chen, M. E. Sanders, A. B. Chakravarthy, Y. Shyr, et al., “Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies,” The Journal of Clinical Investigation, 121: 2750-2767, 2011.

Biographies

Iman Behravan received the B.Sc. and M.Sc. degrees in Electronics Engineering from Shahid Bahonar University of Kerman, Kerman, Iran and University of Birjand, Birjand, Iran, respectively. Now, he is a Ph.D. student in University of Birjand under supervision of Professor Seyed Hamid Zahiri. His research interests include big data analytics, pattern recognition, machine learning, swarm intelligence and soft computing.

Seyed Hamid Zahiri received the B.Sc., M.Sc., and Ph.D. degrees in Electronics Engineering from Sharif University of technology, Tehran, Iran, Tarbiat Modares University, Tehran, Iran and Ferdowsi University of Mashhad, Mashhad, Iran in 1993, 1995 and 2005, respectively. Now, he is a full professor in the department of Electronics Engineering, University of Birjand, Iran. His research interests include pattern recognition, evolutionary algorithms and soft computing.

Seyyed Mohammad Razavi received the B.Sc. degree in Electrical Engineering from Amirkabir University of Technology, Tehran, Iran, in 1994 and the M.Sc. degree in Electrical Engineering from the Tarbiat Modares University, Tehran, Iran, in 1996, and the Ph.D. degree in Electrical Engineering from the Tarbiat Modares University, Tehran, Iran, in 2006. Now, he is an Associate Professor in the Department of Electrical and Computer Engineering, the University of Birjand, Birjand, Iran. His research interests include Computer Vision, Pattern Recognition and Artificial Intelligence Algorithm.

Roberto Trasarti was born in 1979 in Italy. He graduated in Computer Science in 2006, at the University of Pisa. He discussed his thesis on ConQueSt: A constraint-based query system aimed at supporting frequent patterns discovery. He started the Ph.D. in Computer Science at the School for Graduate Studies "Galileo Galilei", (University of Pisa). In June 2010, he received his Ph.D. presenting the thesis entitled "Mastering the spatio-temporal knowledge discovery process". He is currently a member of ISTI-CNR, and also a member of Knowledge Discovery and Delivery Laboratory. His interests regard Data mining, Spatio-Temporal data analysis, Artificial intelligence, and Automatic Reasoning.

How to cite this paper:
I. Behravan, S. H. Zahiri, S. M. Razavi, R. Trasarti, “Clustering a big mobility dataset using an automatic swarm intelligence-based clustering method,” Journal of Electrical and Computer Engineering Innovations, 6(2): 251-271, 2018.

DOI: 10.22061/JECEI.2019.5243.206

URL: http://jecei.sru.ac.ir/article_1117.html