# Fast and Effective Spatial Clustering Using Multi-Start Particle Swarm Optimization Technique

K. Nafees Ahmed[#1], T. Abdul Razak[*2]

[#1] Research Scholar, Department of Computer Science, Jamal Mohamed College,
Bharathidasan University, Tamil Nadu, India
[1] nafeesjmc@gmail.com

[*2] Associate Professor, Department of Computer Science, Jamal Mohamed College,
Bharathidasan University, Tamil Nadu, India
[2] abdul1964@gmail.com

***Abstract*—Increase in digitalization has led to an increase in the data available for digital processing. Such data are termed as rich data, as they depict direct real time data. Difficulty arises when trying to process such data due to their inconsistent nature. This paper presents a density based clustering technique that can be used to identify arbitrary shaped clusters in data. The advantage of this approach is that it requires no external input to identify the range threshold. Particle Swarm Optimization is used as the selection technique to identify nodes belonging to a cluster. A multi-start variant of the Particle Swarm Optimization technique is used, which parallelizes the entire clustering process making it faster and more efficient. Experiments were conducted and it was identified that the current approach exhibits faster clustering process with better efficiency.**

**Keyword-** Density based Clustering, Arbitrary shaped clusters, Metaheuristics, Particle Swarm Optimization, spatial databases, Noise elimination

## I. INTRODUCTION

Numerous applications require effective management and analysis of spatial data. They correspond to real time requirements such as analysis of satellite imagery, X-ray crystallography and several such applications. This makes automated knowledge discovery mandatory. This paper presents a density based clustering algorithm that can be used to group spatial data. The following are the major concerns to be dealt with while constructing algorithms to operate on spatial datasets:

1.The domain knowledge dealing with spatial data would be very low, hence the algorithm should be able to operate in unsupervised manner

2.The groups would be in arbitrary shapes, hence a defined centroid or grids cannot be used

3.Databases would be huge. Since clustering requires operating on all the available data, the operating algorithm should exhibit faster and efficient processing techniques.

Metaheuristics are a class of algorithms that can be used to generate near optimal solutions at a considerably lesser time. The advantage of this approach is that the time taken by metaheuristics is much lesser when compared to statistical techniques (K Means or Partitioning Around Medoids (PAM)), but this occurs at a slight tradeoff in terms of accuracy. An application that can tolerate this slight accuracy tradeoff can be accommodated by metaheuristics effectively. Metaheuristics are the best techniques for applications involving huge data due to their faster capability in solving problems.

This paper presents a density based clustering technique using Multi-Start Particle Swarm Optimization, which is a variant of the regular Particle Swarm Optimization technique. The remainder of this paper is structured as follows; section II provides the related works, section III presents a detailed working of the density based spatial clustering using Multi-Start PSO, section IV presents the results and discusses them and section V concludes the study.

## II. RELATED WORKS

Several contributions providing solutions for spatial clustering exists in literature. This section presents some of the major contributions in this area. DBSCAN [1], one of the mostly used density based clustering techniques was proposed by Ester et al., in 1996. A major advantage of this approach is that DBSCAN relies on the density of the points contained in the cluster, hence this approach effectively identifies arbitrary shaped clusters. It requires a single input parameter (range threshold) for its efficient operation. This value has to be manually provided by the user to begin the operation. OPTICS [2] is a generalization of DBSCAN that avoids the input requirement. The major drawback of DBSCAN and OPTICS is that they require a drop in the data density to identify the range threshold. They cannot detect intrinsic clusters that form the majority of cluster types in real

time data. A parameter quantification method that is used to effectively cluster spatial data is presented in [3]. This technique uses gamma distributions to model squared distance of points to their second nearest neighbors. The major focus of this method lies in identifying the proportion of points lying in the cluster and other cluster properties such as mean cluster size and mean cluster radius without depending on the prior knowledge of the parameters. A DBSCAN based parallel processing technique CudaSCAN is presented in [4]. This method improves the efficiency of DBSCAN by using GPUs to improve the performance and speed of the clustering process. The entire dataset is partitioned into sub-regions and passed to the GPU cores. A similar GPU based clustering method was proposed by Loh et al. in [4]. This method leverages the power of parallelization, but the major downside is that it cannot handle large amounts of data. Each of these sub-regions are clustered locally and finally these clusters are merged to obtain the final set of clusters. Several methods were proposed to improve DBSCAN [7-12]. A density based clustering algorithm considers both attribute likeness and spatial closeness was proposed by Liu et al. in [13]. This method takes into consideration the geometrical properties of spatial objects as the base components for clustering. Delaunay triangulation with edge length constraints is used to identify the spatial proximity relationships. A study that describes a density based algorithm to perform effective clustering is presented by Nanda et al. in [19]. It is a variant of the DBSCAN algorithm. It defines a new merging criterion, considering correlation coefficient as the similarity measure. The drawback of this approach is that it can discover only fixed shape clusters. A Delaunay's Triangulation based clustering method is presented by Deng et al. in [20]. It discovers complex shaped clusters without prior domain knowledge. The major downside of this approach is that it requires application based parameter tuning. Rough-DBSCAN [21] is another DBSCAN based approach proposed by Vishwanath et al. to improvise the speed of the clustering technique. Clustering on streaming data [22] is also another major area that is on the raise. A density based clustering technique that performs parameter reduction and outlier detection is presented by Cassisi et al. in [16]. This method uses the concept of space stratification and can effectively identify clusters with heterogeneous densities. Due to the reduction in the input parameters, the time taken for processing is effectively reduced.

Several real life applications of density based clustering are available to justify the use of such techniques on huge data. DBSCAN was used to cluster genes with similar expressions by Edla and Jana in [5]. A traffic analysis system that was used to identify hot regions in Shanghai was presented by He et al. in [6]. DBSCAN was used on 1.9 billion GPS records to identify hot areas, which helped cops identify faulty behavior. Similar crime identification application of spatial clustering is presented by Estivill-Castro et al. in [14]. Other major utilities include land use detection [15], earthquake analysis [19] and geographic customer segmentation [17]. A real time based clustering system that considers the presence of obstacles or facilitators during the clustering process is presented by Zhao et al. in [18]. The advantage of this approach is that it does not require additional preprocessing and automatically identifies adjacent arbitrary shaped clusters.

## III. Density based Spatial Clustering Using Multi-Start Particle Swarm Optimization Technique

Particle Swarm Optimization (PSO) is a metaheuristic technique used to provide near optimal results for optimization problems. The major advantage of PSO is that it performs optimizations in a time sensitive manner, hence results from PSO can be obtained in real time. The current approach uses a multi-start PSO to perform density based clustering as shown in Fig 1. The presented approach is divided into three major phases namely; identifying cluster thresholds, cluster nodes identification using multi-start PSO and cluster creation.

Density based clustering is the process of identifying nodes belonging to a cluster on the basis of the density of the nodes contained in the specified area. The threshold level for identifying if a point can be contained within a cluster varies within datasets, hence the initial phase performs data analysis to identify this threshold. Two major properties play a vital role in determining the threshold*; maxDist* and *minPts*. *maxDist* corresponds to the maximum distance threshold that can be tolerated to include a node into an available cluster. *minPts* refers to the minimum number of nodes that needs to be surrounded by the current node in order for the node to become a part of a cluster. Nodes $n_1$ and $n_2$ satisfying this property are known as directly density reachable. Both these points constitute within the same cluster.

Many such directly density reachable points constitute a cluster. Nodes in a cluster are considered to be density reachable, meaning; a node $n_1$ in a cluster is density reachable from a node $n_2$ if there exist a chain of directly density reachable points $p_1$, $p_2$,…$p_n$, where $p_1=n_1$ and $p_n=n_2$.

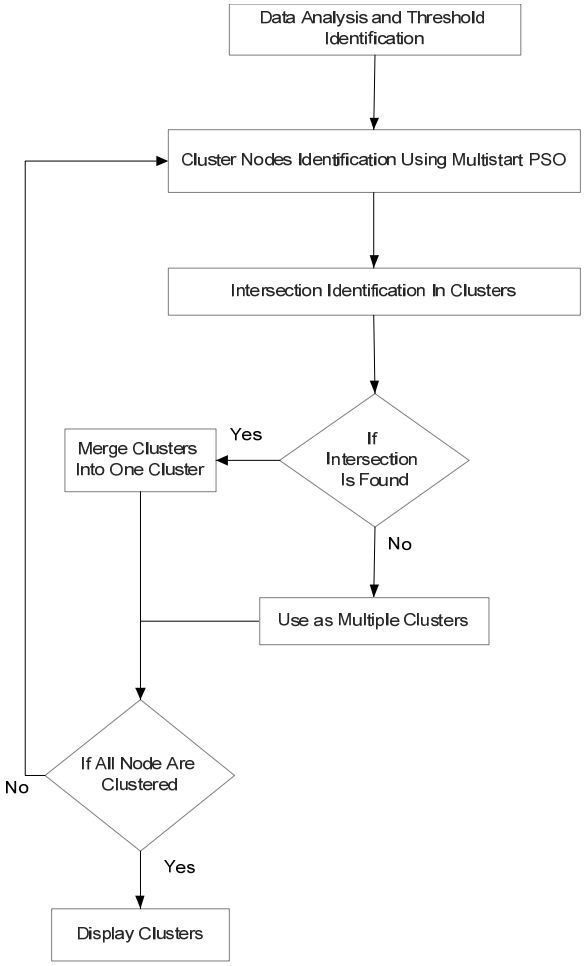


Fig. 1. Density based Spatial Clustering Using Multi-Start Particle Swarm Optimization

Since nodes contained in a cluster are selected based on their neighborhood density, clusters obtained from this method need not be fixed shaped. Hence clusters of any shape can be identified by this method and there is no fixed cluster centroid. All nodes that does not belong to any cluster are defined as noise.

The next phase deals with identifying the clusters using Multi-Start Particle Swarm Optimization (MS-PSO). The first phase of PSO is the initialization of particles to a base node. In our approach, this node corresponds to the reference node from which the other neighboring cluster points are identified. The initial velocity is determined by identifying the search space boundaries ($b_{lo}$ and $b_{up}$).

$$V_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$$

Acceleration is then triggered by applying the velocity on the current particles and by varying their positions. This stage marks the beginning of the particle's acceleration process. The particle's best result (*pbest*) is identified by obtaining the fitness value in the current position of the particle. In the first iteration, *pbest* is always set to the current value. In the subsequent iterations, the current fitness is compared with the *pbest* and the best of the them is selected as the current *pbest*. PSO operates on continuous domain, while the current operating application is in the discrete domain. Hence the continuous values obtained from PSO are discretized to obtain the nearest available node using the following equation

$$P' = \min\left(\sum_{j=1}^{n}\left(\sum_{k=1}^{d}\sqrt{(P_{ik} - N_{jk})^2}\right)\forall i = 1\ to\ p\right)$$

Where $P_{ik}$ refers to the particle *i*'s current location corresponding to dimension *k*, $N_{jk}$ refers to the *k* th dimension of node $N_j$ , d and n corresponds to the total dimensions and total number of nodes respectively.

After the initial movement, the velocity of particles is calculated using the below formula

$$V_{i,d} \leftarrow \omega V_{i,d} + \varphi_p r_p(P_{i,d} - X_{i,d}) + \varphi_g r_g(g_d - X_{i,d})$$

Where $r_p$ and $r_g$ are the random numbers, $P_i,d$ and $g_d$ are the parameter best and the global best values, $x_i,d$ is the value current particle position, and the parameters $\omega$, $\varphi_p$, and $\varphi_g$ are selected by the practitioner.

Every particle follows through the same process and after the identification of each of the *pbest* values, the obtained *pbest* is compared with the current global best solution (*gbest*). If the current pbest has a better fitness compared to the current *gbest* then it is assigned as the new *gbeswt* else the old *gbest* value is retained. In this approach, the fitness of a node is calculated by identifying its *maxDist* from the base node and *minPts* of the node. If both the values obey the threshold condition, then the node is considered as a part of the current cluster. If the *gbest* has satisfied the threshold constraints, it is considered as the base node for the next iteration. This process is repeated until the user set stopping criterion is met. The resultant set of nodes are considered as a cluster. This describes the working process of a single PSO algorithm (shown in Fig 2). The multi-start version of the algorithm operates by running several parallel versions of the PSO algorithm in parallel. Each variant is assigned a different base node. Since they run in parallel, the number of clusters obtained is equal to the number of multi start versions running in the system. Communication is permitted only in the last phase of the process after identifying the clusters. This approach uses five parallel versions of the multi-start algorithm for identifying the results.

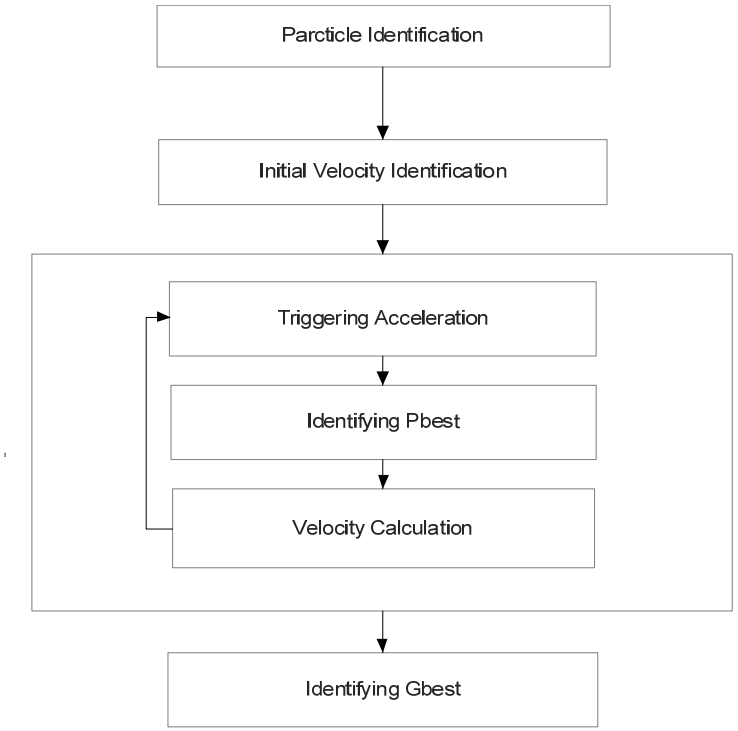


Fig. 2. PSO Working

Though several clusters are obtained after the convergence of the multi-start PSO algorithm, not all clusters are distinct. Some of them overlap each other due to the presence of intersection points. Hence the obtained clusters are checked for intersecting points, and if found, the two clusters are merged to form a single cluster. Repeating values are made eliminated and the distinct values are combined to obtain the initial level clusters.

In the initial iterations, the clusters obtained are considered as the final cluster sets. In the subsequent iterations the initial level clusters are checked with the final cluster sets for any intersecting data. If available, the initial level clusters are combined with the corresponding final cluster sets. In our approach, the termination criterion is met when all the nodes are processed and constitute to be a component of some cluster. After the termination criterion is met, the final cluster sets are examined for any intersections with each other and if available they are merged to form a single cluster. This marks the end of the clustering process.

## IV. RESULTS AND DISCUSSION

Analysis of the proposed approach was conducted by implementing Multi-Start PSO based clustering algorithm on C#.Net. Datasets were obtained from UCI [23] and KEEL [24] repositories. Efficiencies were observed in terms of inter-cluster distance, intra cluster radius, data density contained in each cluster and the time taken by the sequential and parallel variants. Analysis was also considered on the basis of different maxDist values (in terms of percentile division) to identify the best levels of threshold for each data.

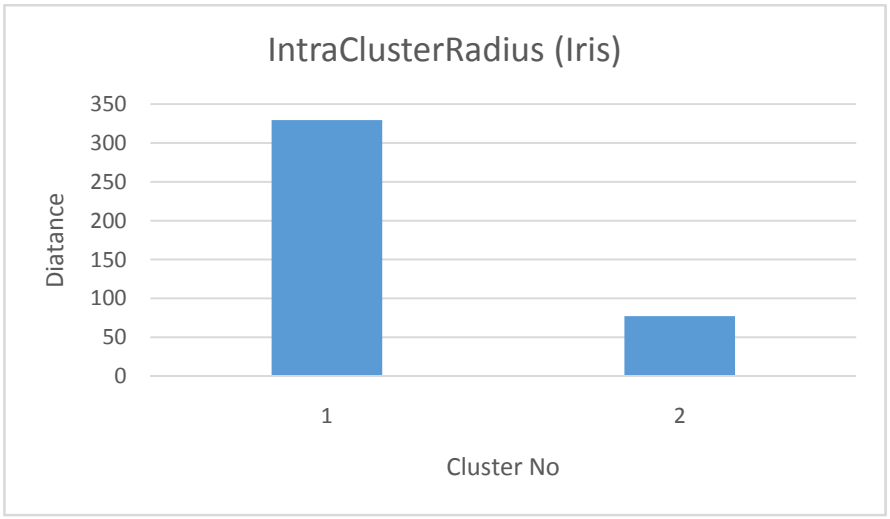


Fig. 3. Intra-Cluster Radius (ICR) – Iris

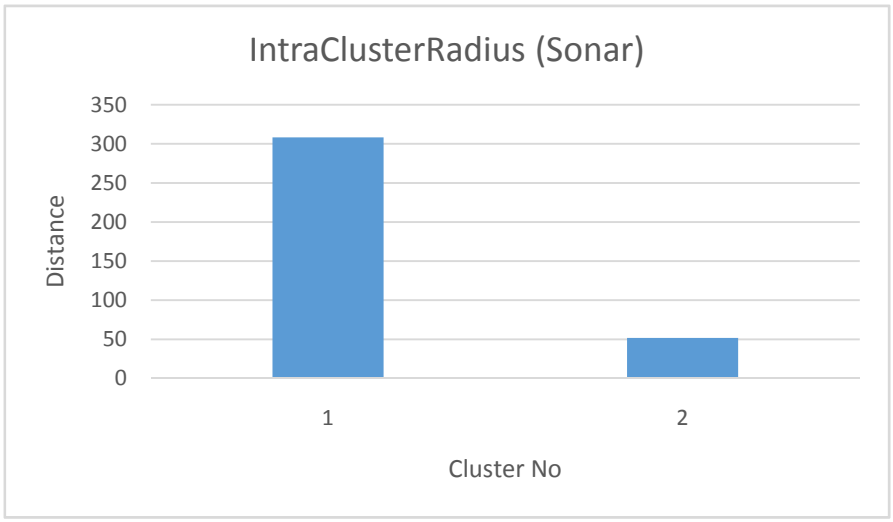


Fig. 4. Intra-Cluster Radius (ICR) - Sonar

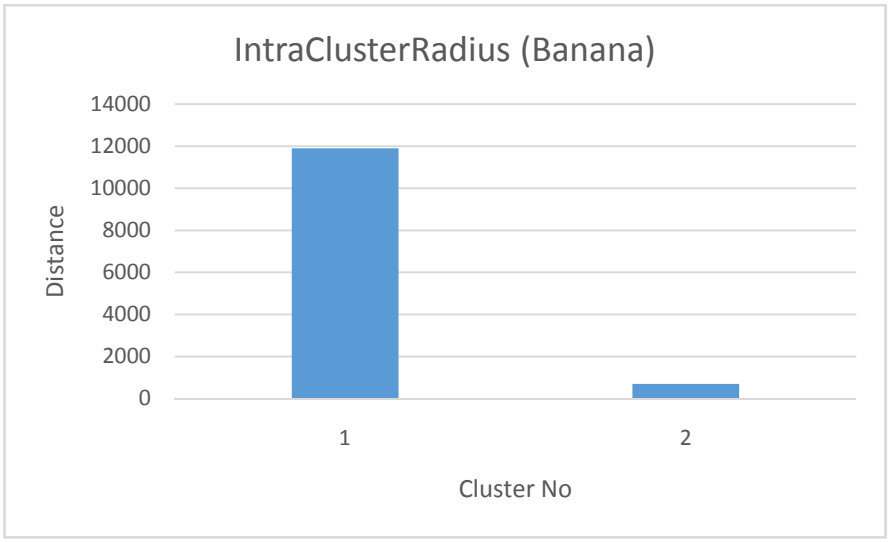


Fig. 5. Intra-Cluster Radius (ICR) – Banana

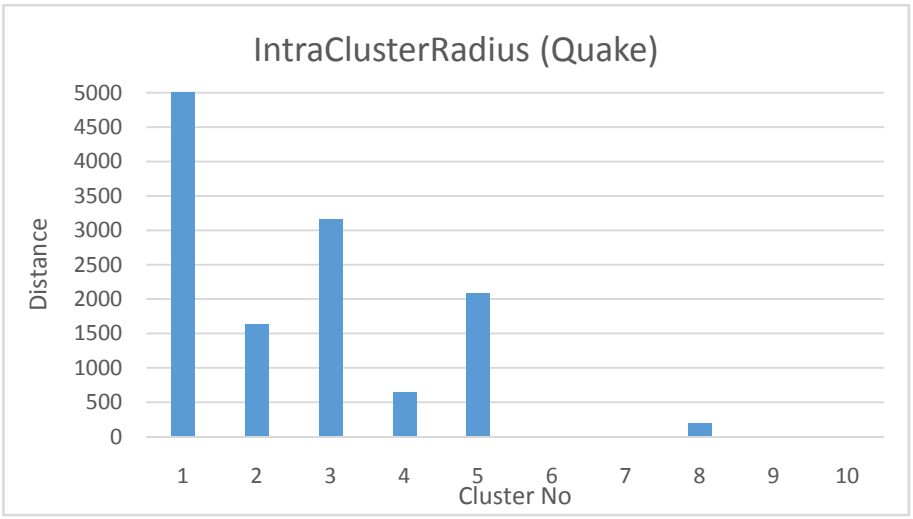


Fig. 6. Intra-Cluster Radius (ICR) – Quake

Figures 3-6 present the intra cluster radius obtained from the data. It was identified that several clusters were obtained and the average intra cluster distances exhibited by the system exhibits effective clustering process. Quake dataset exhibits several clusters with zero intra cluster radius, exhibiting outliers contained in it. Iris, Sonar and Banana datasets exhibit effective clustering processes, exhibiting appropriate cluster groupings.

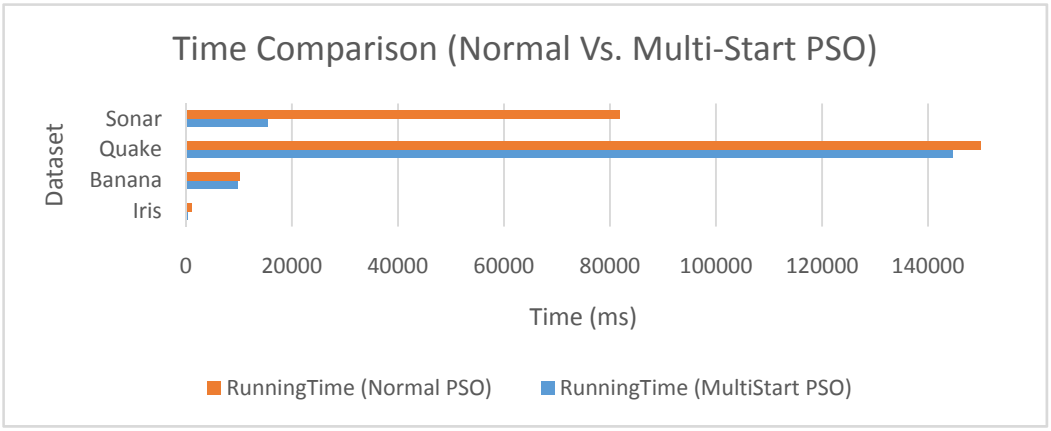


Fig. 7. Time Comparison (Normal Vs. Multi-Start PSO)

Figure 7 present a time comparison between the normal algorithm with the parallel multi-start algorithm. It could be observed that an improvement in time of ~4X to 5X has been observed in several datasets. This exhibits the effectiveness of the multi-start PSO when compared to the sequential PSO.

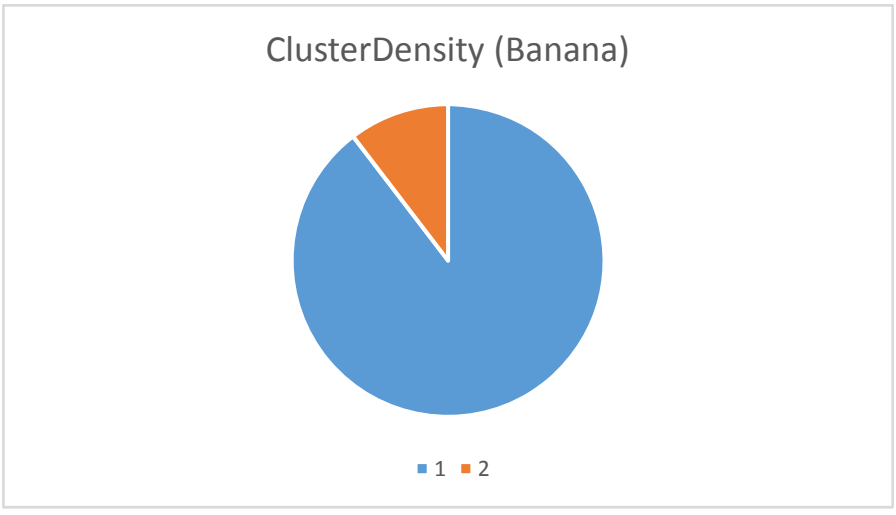


Fig. 8. Cluster Density (Banana)

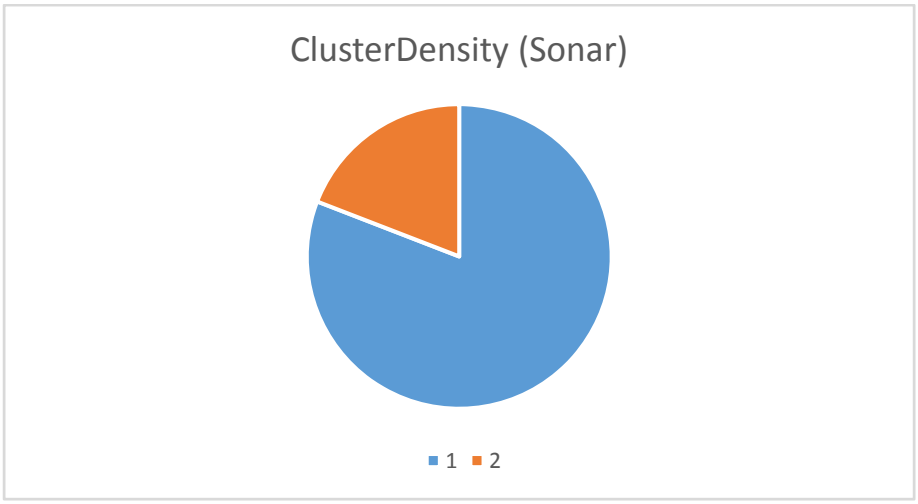


Fig. 9. Cluster Density (Sonar)

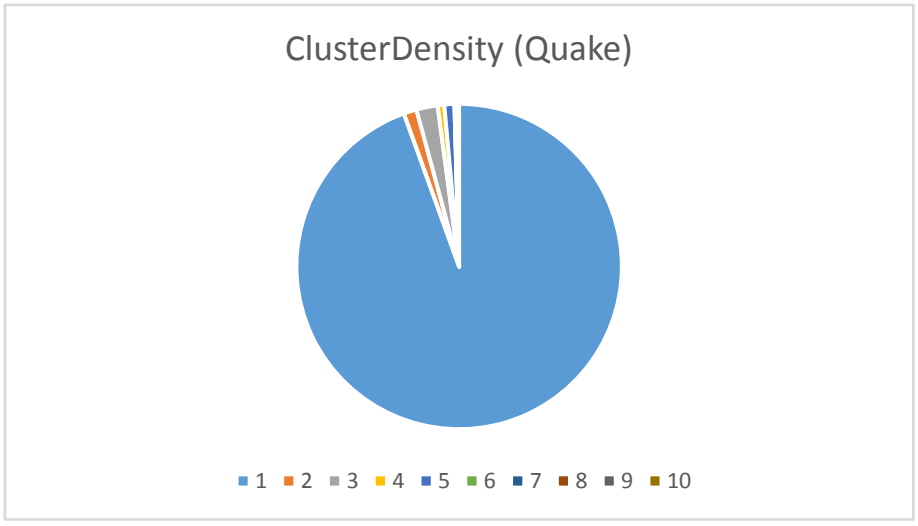


Fig. 10. Cluster Density (Quake)

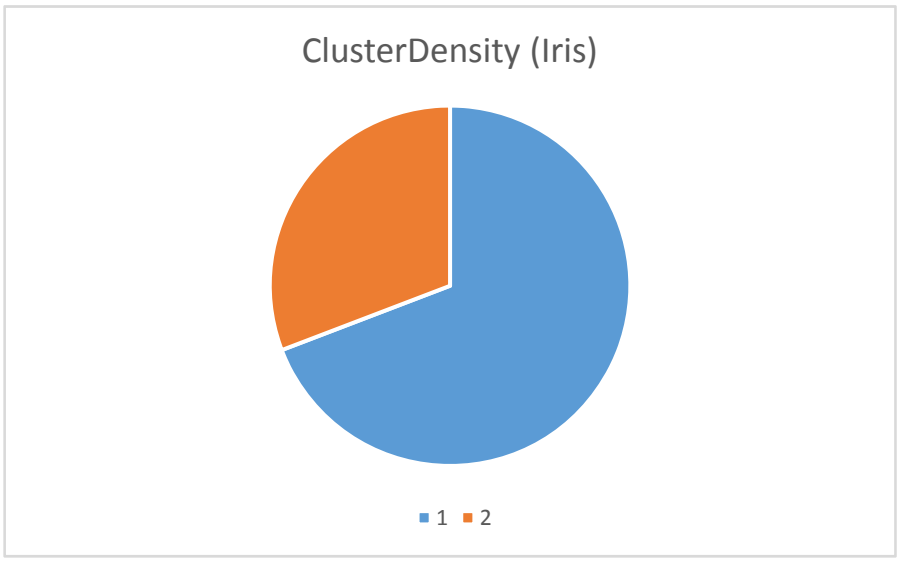


Fig. 11. Cluster Density (Iris)

The figures 8-11 correspond to the density contained in each cluster. It could be observed that the clusters obtained from each of the datasets exhibit varied densities (some low and some very high). Hence it could be validated that the algorithm effectively identifies varied shaped clusters with varied densities. Figure 12 represent the average inter cluster distance on all datasets. It could be observed that effective inter-cluster

distances were observed in all the datasets. This exhibits the ability of the algorithm to effectively identify different shaped clusters.

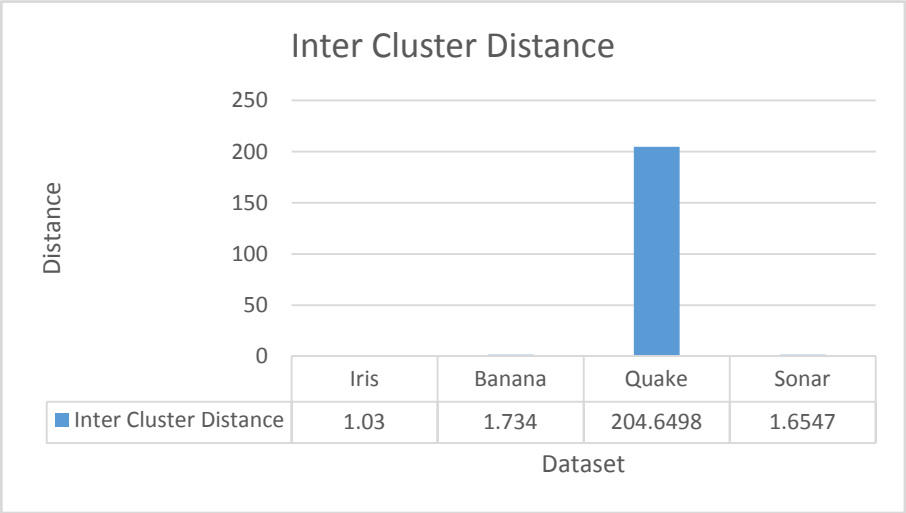


Fig. 12. Inter Cluster Distance

## V. CONCLUSION

A fast and effective algorithm for identifying clusters using PSO based techniques is presented in this paper. The current approach is based on density based clustering performed in parallel using the multi-start PSO. Inclusion of parallelization has provided a huge improvement in terms of time. The advantage of this approach is that, since the density of a point is considered as the base in constructing a cluster, clusters of arbitrary shapes can be created. It does not require prior domain knowledge and has the advantage of operating on huge data. Future research directions include improving the process by utilizing GPU based parallelization rather than CPU based parallel implementations. Further improvements can be made by analyzing and preprocessing the data set to reduce unnecessary data, hence improving the performance.

## REFERENCES

[1] M. Ester,H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise. InKdd 1996 Aug 2 (Vol. 96, No. 34, pp. 226-231).
[2] M. Ankerst, M.M. Breunig, H.P. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure. InACM Sigmod Record 1999 Jun 1 (Vol. 28, No. 2, pp. 49-60). ACM.
[3] M. Schäfer, Y. Radon, T. Klein, S. Herrmann, H. Schwender, P.J. Verveer, K. Ickstadt, A Bayesian mixture model to quantify parameters of spatial clustering. Computational Statistics & Data Analysis. 2015 Dec 31;92:163-76.
[4] W.K. Loh, H. Yu, Fast density-based clustering through dataset partition using graphics processing units. Information Sciences. 2015 Jul 1;308:94-112.
[5] D.R. Edla, P.K. Jana, I.S. Member, A prototype-based modified DBSCAN for gene clustering. Procedia Technology. 2012 Dec 31;6:485-92.
[6] Y. He, H. Tan, W. Luo, H. Mao, D. Ma, S. Feng, J. Fan, Mr-dbscan: An efficient parallel density-based clustering algorithm using mapreduce. InParallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on 2011 Dec 7 (pp. 473-480). IEEE.
[7] G. Andradea, G. Ramosa, D. Madeiraa, R. Sachettoa, R. Ferreirac, L. Rochaa, G-DBSCAN: a GPU accelerated algorithm for density-based clustering, Proc. Comput. Sci. 18 (2013) 369–378.
[8] C. Böhm, R. Noll, C. Plant, B. Wackersreuther, Density-based clustering using graphics processors. InProceedings of the 18th ACM conference on Information and knowledge management 2009 Nov 2 (pp. 661-670). ACM.
[9] S. Brecheisen, H.P. Kriegel, M. Pfeifle, Parallel density-based clustering of complex objects. InAdvances in Knowledge Discovery and Data Mining 2006 Apr 9 (pp. 179-188). Springer Berlin Heidelberg.
[10] A. Hinneburg, H.H. Gabriel, Denclue 2.0: Fast clustering based on kernel density estimation. InAdvances in Intelligent Data Analysis VII 2007 Jan 1 (pp. 70-80). Springer Berlin Heidelberg.
[11] M. Patwary, D. Palsetia, A. Agrawal, W.K. Liao, F. Manne, A. Choudhary, A new scalable parallel DBSCAN algorithm using the disjoint-set data structure. InHigh Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for 2012 Nov 10 (pp. 1-11). IEEE.
[12] X. Xu, J. Jäger, H.P. Kriegel, A fast parallel clustering algorithm for large spatial databases. InHigh Performance Data Mining 2002 Jan 1 (pp. 263-290). Springer US.
[13] Q. Liu, M. Deng, Y. Shi, J. Wang, A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity. Computers & Geosciences. 2012 Sep 30;46:296-309.
[14] V, Estivill-Castro, I. Lee, Multi-level clustering and its visualization for exploratory spatial analysis. GeoInformatica. 2002 Jun 1;6(2):123-52.
[15] J. Sander, M. Ester, H. P. Kriegel, X. Xu, Density-based clustering in spatial database: the algorithm GDBSCAN and it applications. Data Mining and Knowledge Discovery 2 (2), 169–194, 1998.
[16] C. Cassisi, A. Ferro, R. Giugno, G. Pigola, A. Pulvirenti, Enhancing density-based clustering: Parameter reduction and outlier detection. Information Systems. 2013 May 31;38(3):317-30.
[17] H.J. Miller, and J. Han, Geographic Data Mining and Knowledge Discovery, .{CRC} Press. New York, 2009.

[18] Q. Zhao, Y. Shi, Q. Liu, P. Fränti, A grid-growing clustering algorithm for geo-spatial data. Pattern Recognition Letters. 2015 Feb 1;53:77-84.
[19] S.J. Nanda, G. Panda, Design of computationally efficient density-based clustering algorithms. Data & Knowledge Engineering. 2015 Jan 31;95:23-38.
[20] M. Deng, Q, Liu, T, Cheng, Y. Shi, An adaptive spatial clustering algorithm based on Delaunay triangulation. Computers, Environment and Urban Systems. 2011 Jul 31;35(4):320-32.
[21] P.Viswanath, V.S. Babu, Rough-DBSCAN: A fast hybrid density based clustering method for large data sets. Pattern Recognition Letters. 2009 Dec 1;30(16):1477-88.
[22] Q. Tu, J.F. Lu, B.Yuan, J.B.Tang, J.Y. Yang, Density-based hierarchical clustering for streaming data. Pattern Recognition Letters. 2012 Apr 1;33(5):641-5.
[23] http://archive.ics.uci.edu/ml/datasets.html?format=&task=cla&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=table
[24] http://sci2s.ugr.es/keel/datasets.php