A Novel Hybrid Clustering Techniques based on K-Means, PSO and Dynamic Optimization

G.Malini Devi
Asst.Professor
Dept. of CSE,GNITS
Hyderabad-8, India.

M. Seetha, PhD
Professor
Dept. of CSE,GNITS
Hyderabad-8, India.

K.V.N.Sunitha, PhD
Principal
BVRIT Hyderabad
Hyderabad-90, India

ABSTRACT
Clustering is a process for partitioning datasets. This technique is a challenging field of research in which their potential applications pose their own special requirements. K-Means is the most extensively used algorithm to find a partition that minimizes Mean Square Error (MSE) is an exigent task. The Object Function of the K-Means is not convex and hence it may contain local minima. ACO methods are useful in problems that need to find paths to goals. Particle swarm optimization (PSO) is a global optimization algorithm for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. But PSO algorithm suffers from slow convergence near optimal solution. In this paper a new modified sequential clustering approach is proposed, which uses PSO in combination with K-Means & dynamic optimization algorithm for data clustering. This approach overcomes drawbacks of K-means, PSO technique, improves clustering and avoids being trapped in a local optimal solution. It was ascertained that the K-Means, PSO, KPSOK & dynamic optimization algorithms are proposed among these algorithms dynamic optimization results in accurate, robust and better clustering.

General Terms
The proposed techniques shows better results with combined approach with PSO compared to individual techniques implementation.

Keywords
Cluster centroids, K-Means, PSO, KPSOK, dynamic optimization, global optimization.

1. INTRODUCTION
Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. The process of dividing data elements into classes or clusters so that items in the same class are similar to each other, and items in different classes are dissimilar to each other is known as data clustering. Clustering algorithms have been used in data mining and machine learning with UCI machine learning repository with many applications arising from a wide range of problems, including plant and animal ecology, medical imaging, biology, computational biology and bioinformatics, principal component analysis (PCA), sequence analysis, mean shift clustering, and bilateral filtering.

Fast and efficient data clustering algorithms play a significant role in helping users to summarize, effectively navigate and organize the information. Recent studies have shown that partitional clustering algorithms are more appropriate for clustering large datasets. The K-means technique is the most commonly used partitional clustering technique because it can be easily implemented and efficient one in terms of the CPU time. The major difficulty with this technique is that it is sensitive to the selection of the initial k value and may converge to local optima. This paper presents a hybrid Particle Swarm Optimization K-means and (PSO) data clustering algorithm that performs fast data clustering and can avoid being trapped in a local optimal solution as well.

Particle Swarm Optimization (PSO) is a population based algorithm [3]. To achieve a self evolution system in bird flocking or fish schooling behavior, this algorithm can be implemented. Even though the search process is not random it searches for optimum solution. According to the different consequences, it decides the searching path by the fitness function. PSO needs smaller parameters to decide the solution, compared to other evolutionary algorithms. PSO has a stable convergence character with great computational efficiency and is easily implemented. A highly capable evolutionary based clustering method by PSO is provided to find the near optimal solution in search space to trounce the previous problems [4].

Dynamic optimization is a method for adapting the swarm optimizer for dynamic environments. On the basis of outdated information the process consists of causing each companion to reset its record of its best position as the environment changes, to avoid making direction and velocity decisions. Two techniques for initiating this process are focused namely triggered resetting, based on the magnitude of the change in the environment and periodic resetting, based on the iteration count.

To classify the datasets into clusters, three algorithms namely K-MEANS, Particle Swarm Optimization (PSO), KDPSO1, KDPSO2 & KPSO with decreasing w have been used. By taking the values of the datasets, the k-means and KPSO optimization algorithms are implemented and analyzed with different ranges of clustering. By comparing the level of clustering and time taken to form cluster by the five algorithms conclusion can be made which is the best algorithm among them. Each algorithm calculates performance index, i.e. finds cluster centers Fitness, Elapsed time, proximity error**.** KDPSO does better clustering as the particles in KDPSO are gathered very closely compared to the other algorithms because it can find solution for the problems of moving goal. The five data sets namely Iris plant, Zoo, concrete, wine & lenses data set used in the analysis, which are taken from UCI machine repository.

2. K-MEANS TECHNIQUE
This algorithm is the most commonly used partitioned clustering algorithm because it can be easily implemented and is the most efficient one in terms of the execution time. It takes the input parameter k and partitions a set of n objects

into k clusters so that the resulting intra cluster similarity is high but the inter cluster similarity is low [1].Cluster similarity is measured with respect to the mean value of the objects in a cluster, which can be known as the cluster's centroid or center of gravity. Selection of k random objects, in which each object initially represents a cluster mean or center. For the remaining objects, an object is assigned to the cluster to which it is the most similar based on the distance between the object and the cluster mean [10]. It then computes new mean for each cluster [6]. This process continues until the criterion function converges.

2.1 Similarity Metric

In clustering technique, the dataset to be clustered is represented as a set of data objects *X={x1, x2, …., xn}*, where the object *xi* corresponds to a single object and is called the class label to which it belongs. The class label should include proper class to represent the object.The similarity between two data objects needs to be measured in a clustering analysis. There exists two prominent ways to compute the similarity between objects m_p and m_j.

2.2 Minkowski Distance

$$D_n(m_p,m_j)=[\Sigma_{i=1}^{dm}|m_{i,p}-m_{i,j}|^n]^{1/n}$$

For n=2, we obtain the Euclidean distance. To manipulate equivalent threshold distances, considering that the distance ranges will differ according to the dimension number, this technique uses the normalized euclidean distance as the similarity metric of two data objects, *mp* and *mj*, in the vector space[2].

$$d\ (m_p,m_j)= \sqrt{([\Sigma_{i=1}^{dm}|m_{pk}-m_{jk}|^2]/d_m)}$$

where m_p and m_j are two data objects; d_m denotes the dimension number of the search space; m_{pk} and m_{jk} stand for the data objects m_p and m_j's weight values in dimension *k*.

2.3 Fitness

In each iteration, the particle adjusts the centroid vector, position in the search space according to its own experience and those of its neighbors. The average distance between a cluster centroid and a data object is used as the fitness value to evaluate the solution represented by each particle. The fitness value is measured by the equation

$$f=\Sigma_{i=1}^{N_c}\{\Sigma_{j=1}^{P_i}d(o_i,m_{ij})/P_i\}/N_c$$

where m_{ij} denotes the j^{th} data object, which belongs to cluster i; O_i is the centroid vector of $\mathbf{i}^{th}$ cluster; $d(o_i, m_{ij})$ is the distance between data object m_{ij} and the cluster centroid O_i.; P_i stands for the data set, which belongs to cluster C_i; N_c stands for the cluster number.

2.4 Purity

It is known that the K-means algorithm tends to converge faster than other clustering algorithms, but usually with a less accurate clustering result. So, here in our paper our clustering accuracy is measured using purity[5].

$$Purity(C_j)=max(|C_j|_{class=i})/|C_j|$$
$$Overall\ Purity=\Sigma_{j=1}^{k}(|C_j|Purity(C_j))/|D|$$

Where k represents the number of clusters, D represents the size of the dataset, $|C_j|$ is size of the clusters and $|C_j|_{class=i}$ denotes the number of items of class i assigned to cluster j.

2.5 Cluster Centroid

Calculate the cluster centroid vector cj, by using equation as follows.

$$cj=1/n_j\Sigma\ d_j \quad forall d_j \in S_j$$

Where d_j denotes the data object that belong to cluster S_j; c_j stands for the centroid vector; n_j is the number of data objects belong to cluster S_j.

2.6 Membership Function

The equation shows membership function where $u(m_k/z_p)$ is the membership function which quantifies the membership of pattern z_p to cluster k. For K-means algorithm in the project the membership and weight are defined as follows.

$$u(m_k/z_p)=1 \text{ if } d^2(z_p,m_k)=arg\ min_k\{d^2(z_p,m_k)\}$$
$$=0 \text{ otherwise}$$

2.7 Pseudo-code for K-means

1.Inheriting cluster centroid vectors from the datasets.
2.Assigning each data object to the closest cluster centroids.
3.Recalculating the cluster centroid vector cj using the above equation.
4.Repeating step 2 and 3 until the convergence is achieved.

3. PSO TECHNIQUE

PSO is an evolutionary based computation method that performs robust and efficient optimization. It is a population-based optimization technique, where a population is represented as swarm. PSO follows a stochastic optimization method based on swarm intelligence. The basic idea is that each particle represents a likely solution which it updates according to its own experience and that of its neighbors [13]. The PSO algorithm searches in parallel using a group of individuals. Individuals in as swarms, approach to the optimal solution through the previous experience, present velocity and the experience of its neighbors. By adjusting the trajectories of moving points in the multi-dimensional space, the PSO searches the problem domain.

The motion of individual particles for the optimal solution is governed through the interactions. The connectivity between the individual particles is established by the position, velocity , best performance of individuals and their neighbors[9].

3.1 Notations for PSO

The position of the i^{th} particle of a swarm of size n, is represented by the D-dimensional search space where *xi*= (*xi*1, *xi*2, ..., *xiD*)

1. The best previous position (i.e., the position giving the best function value) of the i-th particle is recorded and represented by *pi*= (ρ*i*1, ρ*i*2, ..., ρ*iD*).
2. The position change (velocity) of the i-th particle is *Veli*=(*Veli*1, *Veli*2, ..., *VeliD*) .
3. The position of the best particle of the swarm (i.e., the particle with the smallest function value) is denoted by index *pg*.
4. The particles are then manipulated according to the following equations.

$Vel_{id}(t+1)= \chi\{w\ Vel_{id}\ (t)+c_1\varphi_1[\rho_{id}(t)-x_{id}(t)]+\ c_2\varphi_2\ [\rho_{gd}(t)-x_{id}(t)]\}$
$x_{id}(t+1)= x_{id}(t)+ Vel_{id}(t+1)$
where d=1,2,…..D and i=1,2,…..n.
w: inertia weight=0.72

c1,c2 : positive acceleration constants=1.49
φ_1, φ_2 : random numbers
χ : constriction factor=1.0

4. DYNAMIC PSO

The three types of dynamic optimization techniques, which are implemented with PSO. They are DPSO1, DPSO2 & PSO with decreasing w [7].

4.1 Dynamic PSO1

In DPSO1, they are called dynamic because parameters c1 and c2 are updated during the convergence process. There are initial values for c1 and c2:$c1^0$ and $c2^0$.These values are updated according to the rate of improvement of the population best fitness value along the interactions. In these algorithms if the best fitness value remains the same during t interactions, c1 and c2 decrease by s-10% their original values that is $\mathbf{c1=c1^0(1-0.1s)}$ and $\mathbf{c2=c2^0(1-0.1s)}$, where s-1 is the number of times that c1ec2 were already updated. There are minimum values for c1 and c2 defined as $\mathbf{c1^0(1-0.1\rho)}$and $\mathbf{c2^0(1-0.1\rho)}$ [12].

4.2 Dynamic PSO2

In DPSO2, they are called dynamic because parameters c1 and c2 are updated during the convergence process. Besides c1 and c2 decreasing mechanism there is also an increasing rule. There are initial values for c1 and c2:$c1^0$ and $c2^0$.These values are updated according to the rate of improvement of the population best fitness value along the interactions. In these algorithms if the best fitness value improves during t interactions, c1 and c2 decrease by s*10% their original values that is $\mathbf{c1=c1^0(1+0.1s)}$ and $\mathbf{c2=c2^0(1+0.1s)}$, where s-1 is the number of times that c1ec2 were already updated. There are minimum values for c1 and c2 defined as $\mathbf{c1^0(1+0.1)}$and $\mathbf{c2^0(1+0.1).}$

4.3 PSO with decreasing w

The particles evaluate their positions relative to a goal (fitness) at every iteration, and particles in a local neighborhood share memories of their "best" positions, then use those memories to adjust their own velocities, and thus subsequent positions. The original formula developed by Kennedy and Eberhart was improved by Shi and Eberhart with the introduction of an inertia parameter, ω, that increases the overall performance of PSO. In velocity equation the constant ω is reset for every iteration using the following equation.

$$\omega = \omega_{max} - iter*(\omega_{max} - \omega_{min})/iter_{max}$$

Where ω_{min} , ω_{max} is the range in which ω can vary and $\mathbf{iter_{max}}$ is the maximum number of iterations.

4.4 Dynamism on KPSO

For K-Means+PSO all the three versions are applied on KPSO to form the following algorithms**:**KDPSO1**,** KDPSO2 **&** KPSO with decreasing w. There are three parameters which are considered for comparing the dynamic algorithms namely fitness, elapsed time & proximity error. Different parameters represents different feature and thus all the algorithms try to satisfy these parameter and achieve maximum performance[8].

5. RESULTS AND DISCUSSIONS

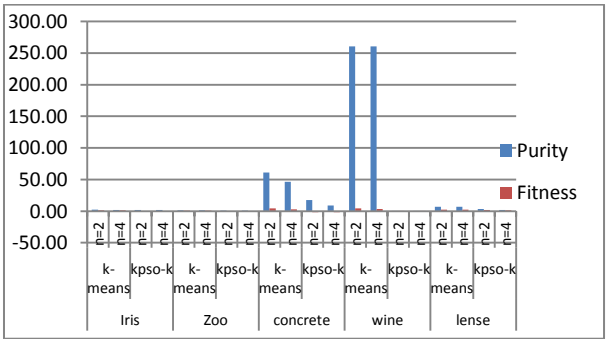
5.1 Datasets description

The Iris plant data set has 150 data points containing 50 instances of each of three types of iris plant[14]. The Zoo dataset has 101 data points, which contain information about an animal in terms of 18 categorical attributes. Each animal data point is classified into 7 classes. The Concrete dataset has 103 data points, which contain information about the types of concrete in terms of 11 categorical attributes. Each concrete data point is classified into 7 classes. The Wine dataset has 178 data points, which contain information about types of wines in terms of 14 categorical attributes. Each wine data point is classified into 5 classes. The Lenses dataset has 24 data points, which contain information about types of lenses in terms of 6 categorical attributes. Each lenses data point is classified into 3 classes. In dynamic optimization, comparison is done with the fitness function, proximity error and CPU time[11].

Table-1:Analysis on the distance measure, purity and fitness for K-Means algorithms:

Dataset	Technique	Distance	Purity	Fitness
Iris	k-means	n=2	2.15	1.56
		n=4	1.85	1.56
	kpso-k	n=2	2.03	-0.83
		n=4	1.67	-1.44
Zoo	k-means	n=2	1.38	0.97
		n=4	1.22	0.97
	kpso-k	n=2	0.97	-0.84
		n=4	0.94	-0.87
Concrete	k-means	n=2	61.03	4.37
		n=4	46.18	3.07
	kpso-k	n=2	17.66	-1.46
		n=4	8.72	-1.47
Wine	k-means	n=2	260.76	4.32
		n=4	260.73	3.37
	kpso-k	n=2	0.41	0.32
		n=4	0.31	0.01
Lenses	k-means	n=2	6.71	2.2
		n=4	6.66	2.2
	kpso-k	n=2	3.16	1.67
		n=4	1.98	1.37

The result table compares the two algorithms, K-Means and KPSO-K in terms of distance measure, purity and fitness value. The conclusion that can be drawn is that, in all the datasets, for n=4, KPSO-K gives low purity and fitness value, thus better performance.



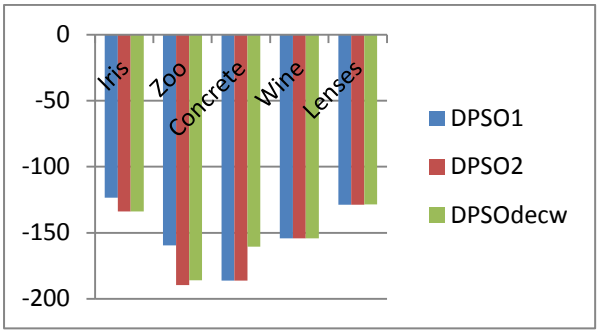
Graph-1: Comparison between the purity and fitness for different distance measures of k-means & kpsok algorithm.

The above graph depicts that the purity and fitness is better for kpsok compare to kmeans because it takes less time for execution.

Table-2: Comparison among different dynamic optimization versions in terms of fitness function with respect to DPSO with five datasets.

Algorithm	Iris	Zoo	Concrete	Wine	Lenses
DPSO1	-123.43	-159.50	-186.25	-154.33	-128.79
DPSO2	-133.85	-189.50	-186.26	-154.33	-128.79
DPSOdecw	-133.85	-185.95	-160.42	-154.33	-128.46

The above results states that for all five datasets, DPSO2 is giving low fitness value with in short span of time.



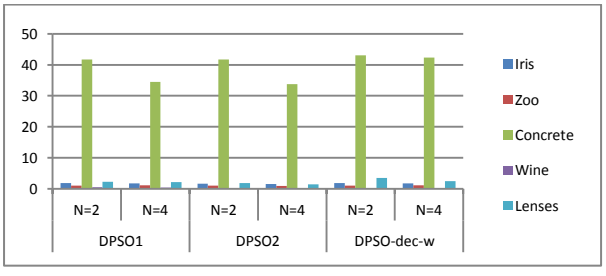
Graph-2: Comparing the dynamic versions based on fitness value with respect to DPSO.

The above graph states that the DPSO2 gives low fitness value because it does not depend on outdated information.

Table-3: Comparison of dynamic optimization in terms of fitness function with respect to k-means & pso with distance measure.

Algorithm	Distance	Iris	Zoo	Concrete	Wine	Lenses
DPSO1	N=2	1.88	1.06	41.73	0.49	2.24
	N=4	1.76	1.08	34.48	0.40	2.12
DPSO2	N=2	1.61	1.04	41.68	0.18	1.85
	N=4	1.59	0.92	33.80	0.04	1.41
DPSO-dec-w	N=2	1.88	1.07	43.03	0.23	3.53
	N=4	1.76	1.09	42.33	0.23	2.42

The tables with respect to DPSO show that for all five datasets, DPSO2 is giving low fitness value.



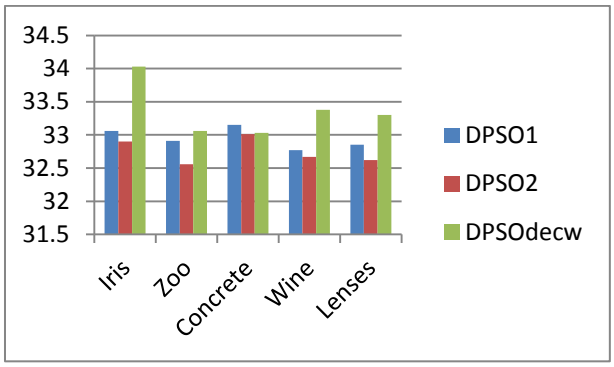
Graph-3: Comparing the dynamic versions based on Fitness value with respect to k-means & pso.

The graph shows that k-means & pso with distance n=4 for DPSO2 give low fitness value because it converges to accurate global optimal solution.

Algorithm	Iris	Zoo	Concrete	Wine	Lenses
DPSO1	33.06	32.91	33.15	32.77	32.85
DPSO2	32.9	32.56	33.01	32.67	32.62
DPSOdecw	34.03	33.06	33.03	33.38	33.3

Table-4: Comparison in different datasets, among the DPSO'S with respect to time factor.

The above table state that time taken by DPSO2 is less compared to the other two versions in all the datasets because the objects can update them so fast when the environment is changed.



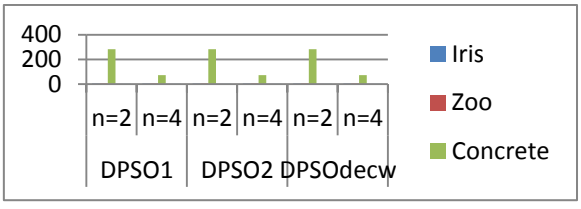
Graph-4: Comparing the dynamic versions based on elapsed time with respect to DPSO.

The above graph states that the DPSO2 takes less time because it can find solution for problems of moving goals.

Table-5: Comparison among different dynamic optimization in terms of time factor with respect to k-means & pso based on distance measure.

Algorithm	Distance	Iris	Zoo	Concrete	Wine	Lenses
DPSO1	n=2	3.42	4.55	281.1	0.6	1.5
	n=4	3.09	4.08	71.54	0.24	1.57
DPSO2	n=2	3.42	4.54	281.1	0.6	1.21
	n=4	3.09	4.07	71.53	0.24	1.12
DPSOdecw	n=2	3.49	4.55	281.1	0.64	2.29
	n=4	3.12	4.09	71.54	0.24	1.57

The above table state that time taken by DPSO2 is less compared to the other two versions in all the datasets.



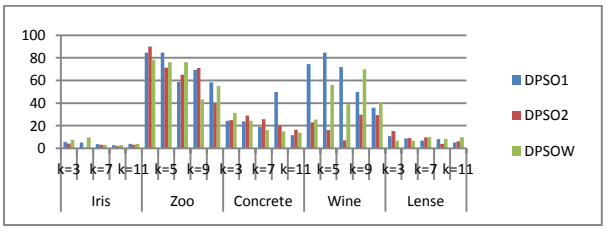
Graph-5: Comparing the dynamic versions based on time factor with respect to k-means & pso.

Dpso2 takes less time compared to other two algorithms because kpso gives best results when dataset are distinct or well separated from each other.

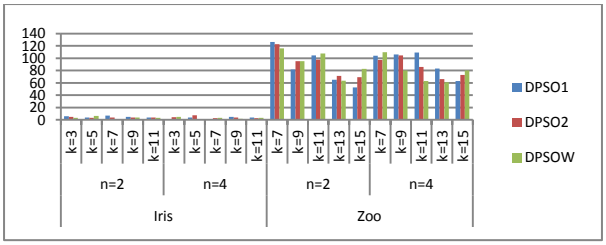
Table-6: Comparison among different Dpso's with respect to proximity error.

Dataset	Clusters	DPSO1	DPSO2	DPSOW
Iris	k=3	5.67	4.25	7.32
	k=5	5.2	0.66	9.78
	k=7	3.78	3.07	3.09
	k=9	2.82	2.14	2.76
	k=11	4.04	3.31	3.94
Zoo	k=3	84.7	90.08	78.34
	k=5	84.65	71.27	76.23
	k=7	58.75	65.07	76.23
	k=9	69.24	70.97	43.25
	k=11	58.31	39.6	55.32
Concrete	k=3	24.34	24.99	31.5
	k=5	23.91	29.03	24.25
	k=7	19	25.92	16.08
	k=9	49.89	19.78	14.94
	k=11	11.61	16.43	14.04
Wine	k=3	74.34	22.92	25.47
	k=5	84.58	16.18	56.11
	k=7	72.02	7.12	39.05
	k=9	49.89	29.83	70.03
	k=11	36.09	29.58	40.46
Lenses	k=3	10.93	15.46	6.88
	k=5	8.94	9.18	6.91
	k=7	6.91	9.7	9.82
	k=9	8.15	4.02	8.19
	k=11	5.01	6.4	9.77

Dpso2 is the best algorithm that results in less number of errors.



Graph-6: Comparing the dynamic versions based on proximity error with respect to DPSO



Graph-7: Comparing the dynamic versions based on proximity error with respect to k-means & pso.

From table-7 & graph-7 proximity error can be considered as a viable and an efficient heuristic to find optimal or near-optimal solutions to clustering problems of allocating N data points to K clusters

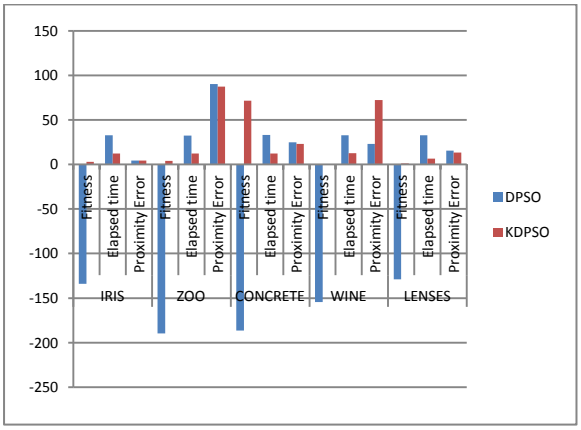
Table-7: Comparison among different dynamic optimization versions in terms of proximity error with respect to k-means & pso based on distance measure.

Dat aset	Distan ce	Clusters	DPSO 1	DPSO 2	DPSOW
Iris	n=2	k=3	6.02	5.1	3.39
		k=5	4.06	3.49	6.63
		k=7	6.72	3.61	0.87
		k=9	4.94	3.8	4.02
		k=11	4.04	4.07	3.55
	n=4	k=3	1.1	4.29	4.81
		k=5	3.98	7.58	0.31
		k=7	1.41	2.94	3.22
		k=9	4.96	3.69	1.76
		k=11	3.9	2.61	3.25
Zoo	n=2	k=7	126.57	122.76	116.11
		k=9	81.53	94.96	95.33
		k=11	104.32	97.95	107.81
		k=13	65.22	71.55	63.51
		k=15	52.79	69.27	82.99
	n=4	k=7	103.93	97.5	109.87
		k=9	106.21	104.56	80.31
		k=11	108.99	85.64	62.88
		k=13	83.05	65.88	61.14
		k=15	63.23	72.9	80.15

Table-8:Comparison on Dpso & Kdpso algorithm

Datasets	Measure	DPSO	KDPSO
IRIS	Fitness	-133.86	3.098
	Elapsed time	32.906	12.3694
	Proximity Error	4.2599	4.2394
ZOO	Fitness	-189.5	4.0707
	Elapsed time	32.56	12.2086
	Proximity Error	90.0898	87.5018
CONCRETE	Fitness	-186.26	71.5311
	Elapsed time	33.0119	12.3102
	Proximity Error	24.9054	22.9505
WINE	Fitness	-154.34	0.24

	Elapsed time	32.679	12.491
	Proximity Error	22.9259	72.1942
LENSES	Fitness	-128.8	1.1287
	Elapsed time	32.6289	6.3518
	Proximity Error	15.4632	13.3752



Graph-8:Result Table for entire dynamic algorithm

The experimental results from graph-8 & table-8 depicts that KDPSO algorithm is best in terms of fitness values, time factor and also in proximity error. For all these three parameters KDPSO gives the least value thus the best dynamic algorithm would be KDPSO.

6. CONCLUSION

Experimental results show that KDPSO2 using distance algorithm for n=4 is the best algorithm. KDPSO2 is best because it is fast, can be applied on larger datasets, gives global optimal solution, and avoids noisy data, outdated information and best performance. In this paper algorithms like K-Means, KPSOK, DPSO1, DPSO2 and PSO with decreasing w are five versions which are implemented on the same metrics. From the results it is concluded that DPSO2 applied on KPSO is the best algorithm when compared to other dynamic algorithms due to minimum intracluster distances as a metric, searches the robust data cluster centers in an N-dimensional Euclidean space.

7. REFERENCES

[1] Xiaohui Huang; Shenzhen Grad. Sch., Yunming Ye ; Haijun Zhang Extensions of Kmeans-Type Algorithms: A New Clustering Framework by Integrating Intracluster Compactness and Intercluster Separation IEEE Transcations on Neural Networks and Learning systems, Volume:25 , Issue: 8, Aug. 2014, pg no1433 – 1446.

[2] ZhouHong-bo , Daqing, China Gao Jun-tao An automatic clustering method based on distance evaluation function-2014 IEEE Workshop on Electronics, Computer and Applications- 2014, page no-10.1109/IWECA.2014.6845701.

[3] Jayshree Ghorpade-Aher, Vishakha Arun Metre, PSO based Multidimensional Data Clustering: A Survey, International Journal of Computer Applications (0975 - 8887),Volume 87 – No.16, February 2014.

[4] M. Imran, R. Hashim, and N. E. A. Khalid, "An overview of particle swarm optimization variants,"Procedia Engineering, vol. 53, pp. 491–496, 2013.

[5] S. C. Satapathy, G. Pradhan, S. Pattnaik, J. V. R. Murthy, and P. V. G. D. P. Reddy, "Performance comparisons of PSO based clustering," InterJRI Computer Science and Networking, vol. 1, no. 1, pp. 18–23, 2009.

[6] Joshua Zhexue Huang,Michael K. Ng, Hongqiang Rong, Zichen Li .Automated Variable Weighting in k-Means Type Clustering[J], IEEE Transactions on Pattern Analysis and Maching Intelligence, 2005, 27(5):657-668.

[7] Chen, Ching-Yi. and Ye, Fun., "Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis," *IEEE ICNSC 2004*, Taipei, Taiwan, R.O.C.,pp. 789_794 (2004).

[8] Van den Bargh, F.; Engelbrecht, A.P. A cooperative approach to particle swarm optimization. *IEEE Trans. Evol. Comp.* 2004, *8*, 225–239.

[9] Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling Multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* 2004, *8*, 240–255.

[10] Chen, Ching-Yi. and Ye, Fun., "K-means Algorithm Based on Particle Swarm Optimization," *2003 International Conference on Informatics, Cybernetics, and Systems*, I-Shou University, Taiwan, R.O.C. pp. 1470□1475 (2003).

[11] Eberhart, R. C. and Shi, Y., "Particle Swarm Optimization:Developments, Applications and Resources," *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2001)*, Seoul, Korea (2001).

[12] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," *Proc. 6th Int. Symposium on Micro Machine and Human Science*, pp.39-43, 1995.

[13] S.Z. Selim, M.A. Ismail, "K-means type algorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Mach. Intell.6*, pp. 81-87, 1984.

[14] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks. 1995. pp. 1942–1948.