International Journal of Computing and Corporate Research

ISSN (Online) : 2249-054X

Volume 4 Issue 1 January 2014

International Manuscript ID : 2249054XV4I1012014-04

# EMPIRICAL EVALUATION OF BBBC AND PSO ALGORITHM FOR DATA CLUSTERING

*Ms. Poonam and Ms. Neelam Oberoi*
*Maharshi Markandeshwar University*
*Saddopur, Ambala, Haryana, India*

***Abstract-***With the rapid development of information technology and database technology, large amounts of data is accessed and stored regularly. The traditional data analysis techniques are not sufficient to extract the inherent relationship between the data and the underlying information in it. There is an urgent need of such algorithms that can intelligently and automatically analyze and transform the data into useful information and knowledge. In this paper we have analyzed the performance of a new optimization algorithm namely Big Bang Big Crunch (BBBC) algorithm on data clustering. We have compared the performance of this algorithm with already used algorithm viz. Particle Swarm Optimization (PSO) technique.

## I INTRODUCTION

Cluster analysis categorizes the data according to maximum similarity and minimum dissimilarity principle. A cluster consists of objects that are similar between themselves and dissimilar to objects of others. It helps in managing and analyzing huge amount of data by clustering similar looking data into one cluster. Clustering is also easy to observe the contents of the organization into hierarchical structure to organize similar events together. Several Data mining techniques and methods such as Statistical Methods, Decision Tree, Neural Network, Genetic Algorithm, Rough Set and Fuzzy Set etc. are used in the main related disciplines and technologies. The main requirements that a clustering algorithm must have is scalability, ability to deal with noise and outliers, insensitivity to order of input records, high dimensionality, interpretability and usability.

Traditionally clustering techniques are broadly divided in hierarchical and partitioning. While hierarchical algorithms build clusters gradually, partitioning algorithms learn clusters directly. In this paper we

will discuss two metaheuristic based algorithm namely BBBC and PSO for clustering purpose.

## II LITERATURE REVIEW

Only the *K*-means algorithm [Zhao2006] and its ANN equivalent, the Kohonen net [Kohonen1990] have been applied on large data sets; other approaches have been tested, typically, on small data sets. This is because obtaining suitable learning/ control parameters for stochastic algorithm is difficult and their execution times are very high for large data sets. The main drawback of well known K-means clustering algorithm is that the cluster result is sensitive to the selection of the initial cluster centroids and may converge to the local optimal and it generally requires a prior knowledge of the probable number of clusters for a data collection [Zhong2005].

Several Evolutionary techniques such as genetic algorithms (GAs), Simulated annealing (SA) and PSO has been used to the data clustering problem [Alwee2009, Sherin2007 and Mariam2013]. The major drawback is that the number of cluster is initially unknown and the clustering result is sensitive to the selection of the initial cluster centroids and may converge to the local optima. A new metaheuristic approach Big Bang Big Crunch has been tested for the purpose of clustering by Hatamalou but has not been compared with other similar looking algorithm such as PSO technique till now for the same set of test set. [Hatamlou2011] So our major theme in this paper will be to implement PSO and BBBC based cluster techniques and then compare both of these algorithms for efficiency analysis.

## III METHODOLOGY

### 3.1 Particle Swarm optimization (PSO)

PSO is a population-based a biologically inspired algorithm which applies to concept of social interaction to problem solving where each individual is referred to as particle and represents a candidate solution. Each particle in PSO flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also flying experience of other particles using the following equations.

$$v_i^d(t+1) = w \times v_i^d(t) + \varphi_1 \times rnd() \times (p_i^d - x_i^d(t)) + \varphi_2 \times rnd() \times (p_g^d - x_i^d(t)) \text{-----} (1)$$

International Journal of Computing and Corporate Research

ISSN (Online) : 2249-054X

Volume 4 Issue 1 January 2014

International Manuscript ID : 2249054XV4I1012014-04

$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$----------
(2)
where

- $v_i^d(t+1)$ is a velocity vector at t+1 time for i particle in $\boldsymbol{d}$ dimension
- . $x_i^d(t+1)$ position vector at t+1 time for i particle in $\boldsymbol{d}$ dimension
- $\boldsymbol{rnd}(\,)$ is random number generator.
- $\varphi_1$ and $\varphi_2$are are learning rates governing the cognition and social components.
- $\boldsymbol{g}$ represents the index of particle with best p-fitness.
- $\boldsymbol{w}$ is the inertia factor that dynamically adjust the velocities of particles gradually focusing the PSO into a local search[Yuhui2005].

Following steps illustrate the overall optimization scheme of PSO in the figure1.

1. Initialize the particle population by randomly assigning locations (X-vector for each particle) and velocities (V-vector with random or zero velocities- in our case it is initialized with zero vector)
2. Evaluate the fitness of the individual particle and record the best fitness $P_{best}$ for each particle till now and update P-vector related to each $P_{best}$.
3. Also find out the individuals' highest fitness $G_{best}$ and record corresponding position $p_g$.
4. Modify velocities based on $P_{best}$ and $G_{best}$ position using eq3.
5. Update the particles position using eq4.
6. Terminate if the condition is met
7. Go to Step 2

**Figure 1** PSO Algorithm

### 3.2 Big Bang and Big Crunch Search Algorithm

The Big Bang and Big Crunch theory is introduced by Erol and Eksin [Erol2006]. The method of optimization stems from an initial population, or universe, which is reduced as the bodies (or solutions) are attracted by other bodies with bigger matter (or bigger aptitude), either because of the matters or the relative distances. The process is concluded when a single body remains in the space, which will is supposed to constitute the optimal solution. Below in figure 2, algorithm for the BBBC algorithm in steps is given.

International Journal of Computing and Corporate Research

ISSN (Online) : 2249-054X

Volume 4 Issue 1 January 2014

International Manuscript ID : 2249054XV4I1012014-04

1. Create random population of solution.
2. Evaluate Solutions.
3. The fittest individual can be selected as the center of mass.
4. Calculate new candidates around the center of mass by adding or subtracting a normal random number whose value decreases as the iterations elapse.
5. The algorithm continues until predefined stopping criteria has been met.

**Figure 2.** BBBC Search Algorithm Workflow

### 3.3 Data Clustering

Each particle maintains a matrix $X_i = (C_1, C_2, \ldots, C_i, .., C_k)$, where $C_i$ represents the $i^{th}$ cluster centroid vector and k represent the total number of clusters. According to its own experience and those of its neighbors, the particle adjusts the centroid vector position in the vector space at each generation. The average distance of data objects to the cluster centroid will be used as the fitness value to evaluate the solution represented by each particle. The fitness value will be measured by the Euclidian Distance between different dimensions of Center of cluster and instances.

## IV EXPERIMENTATION & RESULTS

For clustering, we have used normalized data with Euclidian Distance based fitness function for adjudging the quality of cluster in clustering problem. The clustering problem used for the purpose of this thesis is ***Iris plants database***: This is a well-understood database with 4 inputs, 3 classes and 150 data vectors.

### 4.1 Fitness function

We have used Euclidian fitness function for our experimentation. The Euclidean distance is the straight-line distance between two pixels. For two pixel points or two data points $(x_1, y_1)$ & $(x_2, y_2)$, the Euclidean distance is

- Euclidean distance = $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, ----------(3)

### 4.2 Metric for cluster Analysis

We have used Jaccard index for evaluating the quality of clustering. The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1. An index of 1 means that the two dataset are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula:

$J(A,B) = \frac{A \cap B}{A \cup B}$----------(4)

This is simply the number of unique elements common to both sets divided by the total number of

International Journal of Computing and Corporate Research

ISSN (Online) : 2249-054X

Volume 4 Issue 1 January 2014

International Manuscript ID : 2249054XV4I1012014-04

unique elements in both sets. This index is used to determine silhouette value. The silhouette value for each point is a measure of how similar that point is to points in its own cluster vs. points in other clusters, and ranges from -1 to +1. It is defined as

$$S(i) = \frac{min(d_{inter\_avg}(i,k)) - d_{intra\_avg}(i)}{max(d_{intra_{avg}}(i)) - \min(d_{inter_{avg}}(i,k))} \quad \text{----------(5)}$$

## V RESULTS AND DISCUSSIONS

Following table shows results of experiments. 100 attempts have been made to record the result of K-means, PSO and BBBC based clustering algorithms. We have recorded best result of 100 attempts. Average silhouette values of these are also shown in the table.

**Table 1** Clustering Result of three Algorithms

| **Clustering Algorithm** | **Best silhouette values** | **Average silhouette values** | **Best Fitness recorded as sum of min distance of each test point from its cluster center** |
|---|---|---|---|
| K-means | 0.6351 | 0.3562 | 25352 |
| PSO | 0.7252 | 0.7083 | 8046 |
| BBBC | 0.7826 | 0.6154 | 14644 |

From table 1 it can be clearly stated that K-means is not a good choice for clustering algorithm especially if data is multi dimensional. Although all of these algorithms are stochastic giving different results at each attempt but averaging of 100 attempts clearly shows that K-mean doesn't perform well in comparison with other two algorithms.

International Journal of Computing and Corporate Research

ISSN (Online) : 2249-054X

Volume 4 Issue 1 January 2014

International Manuscript ID : 2249054XV4I1012014-04

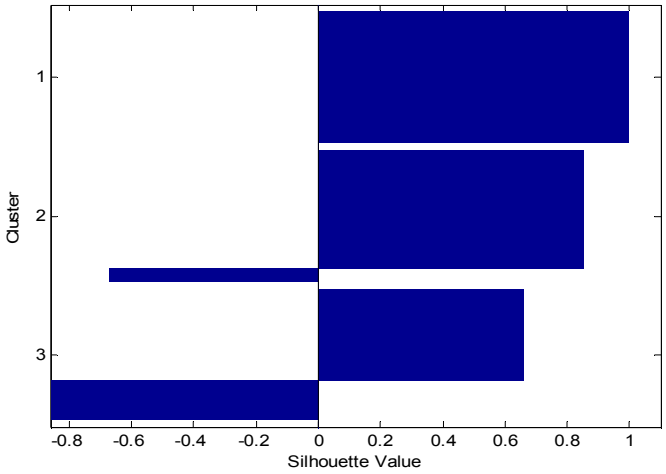


**Figure 3** Silhouette Plot for K-means with Average Silhouette value of 0.6351

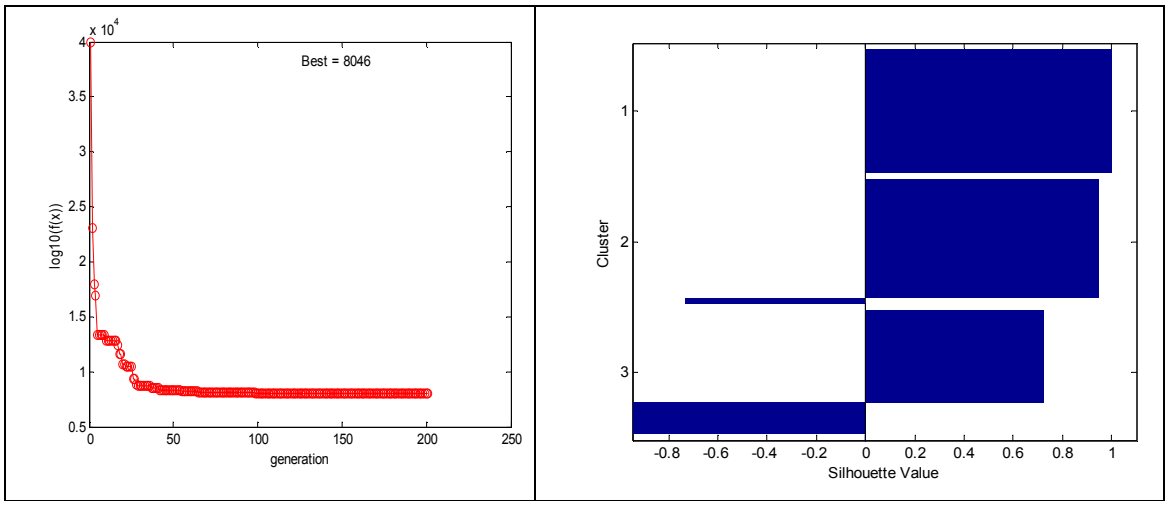


**Figure4** Silhouette Plot for PSO Algorithm with Average Silhouette value of 0.7223

If we compare other PSO and BBBC based algorithms, PSO outperforms BBBC algorithm in averaging result by a huge margin but BBBC has best Silhouette value of 100 attempts by a small margin only. It may suggest that the BBBC is not very consistent in its results. This can also be gauged by comparing figure 5 with figure 6.

The figure 5 suggests a smooth convergence while figure 6 shows that the BBBC algorithm explores

International Journal of Computing and Corporate Research

ISSN (Online) : 2249-054X

Volume 4 Issue 1 January 2014

International Manuscript ID : 2249054XV4I1012014-04

random solution more than PSO algorithm. We may interpret from this discussion that the BBBC algorithm may provide a better global exploration but doesn't provide a smooth convergence suggesting less capable of local exploration.

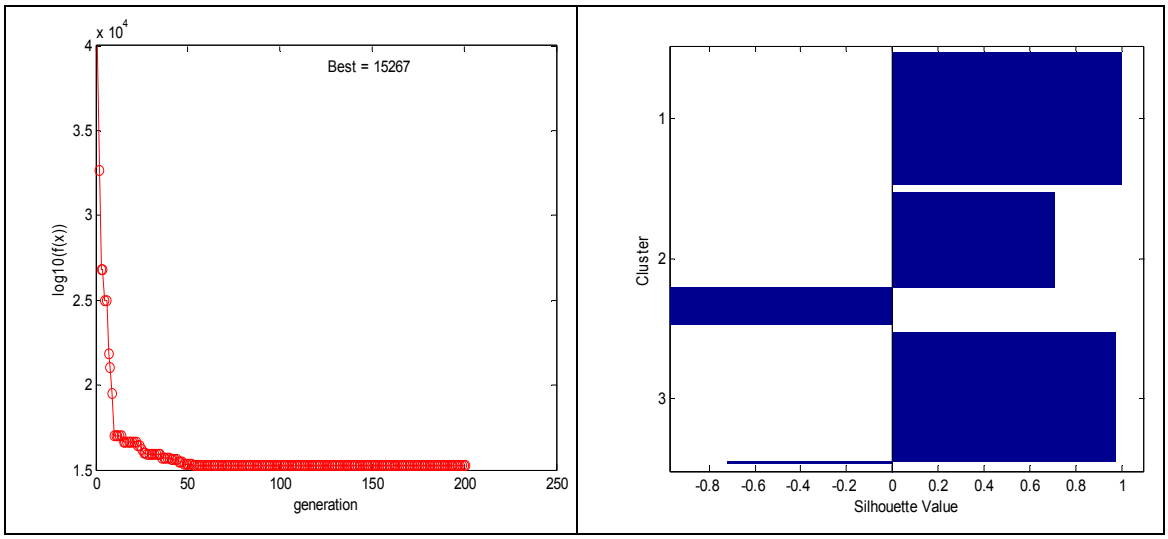


**Figure 5** Silhouette Plot for PSO Algorithm with Average Silhouette value of 0.7252

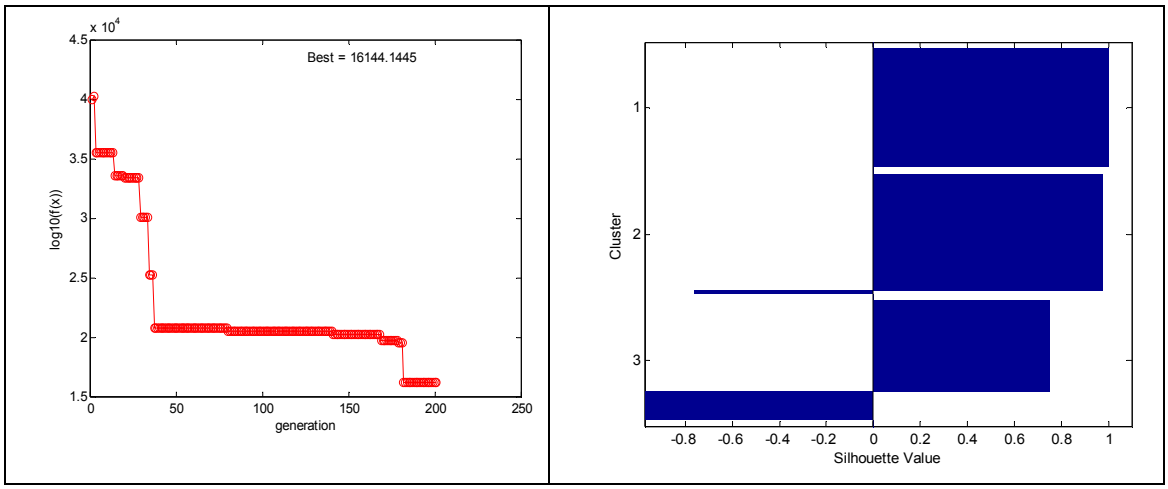


**Figure 6** Silhouette Plot for BBBC Algorithm with Average Silhouette value of 0.7583

ISSN (Online) : 2249-054X

Volume 4 Issue 1 January 2014

International Manuscript ID : 2249054XV4I1012014-04

One more observation we have made that particle fitness is not fully correlated to silhouette value. In Figure 4 the best fitness is 8046 while silhouette value is 0.7223 on the other side In Figure 5 the best fitness is 15267 while silhouette value is 0.7252 on the other side. Similar interpretation can made from figure 6. But this may be due to problem specific. More experiments need to be carried out before generalization of statement.

## VI CONCLUSION

This paper has evaluated BBBC algorithm for data clustering. In comparison to PSO algorithm results have found that PSO performs better than BBBC and have less consistency in results. One more observation is made that particle fitness is not fully correlated to silhouette value. But before generalizing the conclusion algorithm needs to applied for high dimensionality data.

## REFERENCES

[Alwee2009] Razan Alwee, Siti Mariyam, Firdaus Aziz, K.H.Chey, Haza Nuzly, "The Impact of Social Network Structure in Particle Swarm Optimization for Classification Problems". International Journal of Soft Computing , Vol. 4, No. 4, 2009, pp:151-156.

[Eberhart2001] Eberhart, R.C., & Shi, Y. Particle Swarm Optimization: Developments, Applications and Resources. *Congress on Evolutionary Computation*, Seoul, Korea, 81-86. 2001

[Hatamlou2011] Abdolreza Hatamlou, Salwani Abdullah, Masumeh Hatamlou, Data Clustering Using Big Bang–Big Crunch Algorithm, Innovative Computing Technology Communications in Computer and Information Science Volume 241, 2011, pp 383-388

[Mariam2013] Mariam El-Tarabily, Rehab Abdel-Kader, Mahmoud Marie, Gamal Abdel-Azeem, "A PSO-Based Subtractive Data Clustering Algorithm". *International Journal of Research in Computer Science*, 3 [2]: pp. 1-9, March 2013. doi: 10.7815/ijorcs. 32.2013.060

[Pave2002] Pavel Berkhin, "Survey of clustering data mining techniques". Accrue

Software Research Paper, pp.25-

[Sherin2007] Sherin M. Youssef, Mohamed Rizk, Mohamed El- Sherif, "Dynamically Adaptive Data Clustering Using Intelligent Swarm-like Agents". International Journal of Mathematics and Computer in simulation, Vol. 1, No.2, 2007.

[Yuhui1998] Yuhui Shi, Russell C. Eberhart, "Parameter Selection in Particle Swarm Optimization". The 7th Annual Conference on Evolutionary Programming, San Diego, pp. pp 591-600, 1998. doi: 10.1007/BFb0040810

[Zhao2006] Zhao, Tong, Nehorai, Arye, and Porat, Boaz. "K-Means Clustering-Based Data Detection and Symbol-Timing Recovery for Burst-Mode Optical Receiver." *IEEE Transactions on Communications*. Vol. 54. No 8. August 2006. 1492-1501.

[Zhong2005] Zhong Wei, et al. "Improved K-Means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common Structural Property." *IEEE Transactions on Nanobioscience*. Vol. 4. No. 3. September 2005. 255-265.