INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2016; 6(4):569-580

GROUND MOTION CLUSTERING BY A HYBRID K-MEANS AND COLLIDING BODIES OPTIMIZATION

M. Shahrouzi[*, †] and M. Rashidi Moghadam
Department of Engineering, Kharazmi University, Tehran, Iran

ABSTRACT

Stochastic nature of earthquake has raised a challenge for engineers to choose which record for their analyses. Clustering is offered as a solution for such a data mining problem to automatically distinguish between ground motion records based on similarities in the corresponding seismic attributes. The present work formulates an optimization problem to seek for the best clustering measures. In order to solve this problem, the well-known K-means algorithm and colliding bodies optimization are employed. The latter acts like a parameter-less meta-heuristic while the former provides strong intensification. Consequently, a hybrid algorithm is proposed by combining features of both the algorithms to enhance the search and avoid premature convergence. Numerical simulations show competative performance of the proposed method in the treated example of optimal ground motion clustering; regarding global optimization and quality of final solutions.

Keywords: clustering; silhuette; K-means; colliding bodies optimization.

Received: 20 February 2016; Accepted: 23 April 2016

1. INTRODUCTION

Dealing with large or non-organaized data is a challenging task for various fields including earthquake engineeirng. The related methods are formally called *Data Mining* [1]. It includes pattern recongnition, classification and clustering [2]. Clustering is an unspervised method of categorizing a data bank to several clusters so that each entity has the most similarity with its own cluster members and the least with the others [1-3]. Some of the steps usually considered for a typical clustering are: feature extraction or selection, definition of a pattern proximity measure, grouping into clusters and assessment of output [4]. Definition of features may differ case by case while some more common measures have already been presented in literature. A good clustering aims to find a group of clusters on a given data

[*]Corresponding author: Department of Engineering, Kharazmi University, Tehran, Iran
[†]E-mail address: shahruzi@khu.ac.ir (M. Shahrouzi)

which lead to the best measure; therefore it can be considered an optimization task.

Each attribute or feature in the given data matrix may differ from the other ones regarding the type and method of assessment. Such attributes address geotechnical and seismic data in earthquake engineering and the related applications [5, 6]. Sampling methods are of interest to deal with this class of optimization problems. In this regard, metaheuristics can be mentioned since they mostly search the design space by extensively sampling it to find the global optima. They include single-agent methods such as simulated annealing [7, 8] and multi-agent methods including genetic and evolutionary algorithms[9], nature inspired and swarm-based algorithms [10-13], human and culture inspired algorithms [13] and physics-based algorithms [12, 14].

In the present work, a hybrid method is developed based on combination of two algorithms: *K-means* as the widely-used deterministic algorithm for clustering and *Colliding Bodies Optimization*, CBO as a recently developed meta-heuristic [14-17]. This way, both exploraive features and local intensification of these algorithms are combined in the proposed method; called ECBO_KM. The method is further utilized for optimal clustering of ground motion and applied to a database of real-world earthquakes. Quality of results is compared applying three aforementioned methods with variation in the number of clusters. Finally, their performance is compared via numerical simulation to declare effect of the proposed hybridization in the treated clustering problem.

2. OPTIMAL CLUSTERING FORMULATION

Let the database of concern be presented as a matrix with different attributes in its columns where each row of the matrix corresponts to an individual entity. For a data matrix with N rows and M columns, it is possible to distribute such entities into K clusters provided that K is smaller than N. Hence, different ways of dividing the data to a given number of clusters exist that construct the search space.

Every such clustering differ from the others specially in view of a measuring function. The optimization problem can thus be formulated to seek a clustering having the best measure. Silhuette value as a common cost measure is widely used to evaluate quality of a clustering on a given database [18]. It explains how similiar is an entity to its own cluster compared to the other clusters. As defined by the following relation it varies between -1 for the worst case to 1 for the best clustering.

$$s(e) = \frac{b(\mathrm{e}) - a(\mathrm{e})}{\max\{a(\mathrm{e}), b(\mathrm{e})\}} \tag{1}$$

in which $a(e)$ stands for the mean distance of entity, e, to all entities of its own cluster whereas $b(e)$ denotes mean distance of e to the members in the other culsters. A *silhuette plot* is thus obtained by plotting $s(e_i)$ of all e_i entities in the i^{th} cluster and then the next one provided that they are sorted in decending order of their silhuette value[18].

Therefore, silhuette value of each entity shows its similarity to its own cluster together with its dissimilarity to the others. In this regard a fitness function, F, is defined based on

sum of cluster-mean silhuette values over all clusters of the database. Problem formulation in the present research is to maximize the following fitness function:

$$Maximize \;\; F(X) = (1 + r_p \left| 1 - \frac{q}{K} \right|^2) \times \frac{Q}{K} \tag{2}$$

where $Q = \sum_{j=1}^{K} \sum_{i=1}^{N_i} s(e_i)$. Any component of the design vector $X = \langle x_1, ..., x_{Ne} \rangle'$ may be associated an integer number between 1 and *K* to identify its cluster number while N_e is the number of rows in the earthquake clustering data matrix.
N_i is the number of entities in the i^{th} clusters when there is *K* clusters. The first term in the parentheses consitiues a penalty for undesired solutions with *q* clusters rather than *K* ones. It is indeed designated to force the search to generate solutions including a fixed number of *K* clusters. r_p stands for the prescribed penalty coefficient; say in the order of 10.

3. UTILIZED ALGORITHMS FOR CLUSTERING

3.1 K-means algorithm

K-means is a popular method of clustering which is also referred to as Lloyd's algorithm [20]. It aims to quantize a signal set or partition a set of *n* observations, X_j, $j = 1, ..., n$, into *K* clusters, $\{C_1, ..., C_K\}$, so that the sum over distances of each cluster's members with respect to their mean be minimized for the entire data. In another word, the algorithm solves the following problem.

$$\min \;\; f(\{C_1, ..., C_K\}) = \sum_{i=1}^{K} WCSS_i \tag{3}$$

inwhich *Within-Cluster Sum of Squares*, WCSS, for every i^{th} cluster is calculated as:

$$WCSS_i = \sum_{X \in C_i} \left\| X - \mu_i \right\|^2 \tag{4}$$

Starting from a set of prescribed centers, *K-means* works in an iterative manner to improve their positions and the *K* cluster sets, to achive the problem objective. The standard *K-means* algorithm starts by a given set of means $\{\mu_1, ..., \mu_K\}^{(1)}$ and then iteratively switches between the following two steps:

Assignment Step: every obesrvation X_j is assigned to the nearest mean based on the least Euclidean space; $d_{X_j, \mu_i} = \sqrt{\left\| X_j - \mu_i \right\|^2}$.

Repair Step: Update position of every cluster's mean based on its members positions

$$\mu_i = \underset{j}{mean}(X_j), \;\; X_j \in C_i \tag{5}$$

The above steps are repeated until either of the following termination condition is satisfied:

1. Iterations of the algorithm reach a prescribed number.
2. Variation in position of every cluster mean during the last two iterations is negligible.
3. No further membership changes in the clusters occur

It is worth mentioning that although *K-means* is a straight-forward deterministic algorithm, it suffers from dependency to initial guess of means and sensitive efficiency to the size of data matrix [23].

3.2 Colliding bodies optimization

Colliding Bodies Optimization, CBO, is a recent meta-heuristic algorithm represented by Kaveh and Mahdavi [14, 15]. In this method, one object collides with the other object and they move towards a minimum energy level. The CBO does not rely on any internal parameter and remarkably is simple. Every i^{th} colliding body, CB_k, has a specified mass that is calculated as:

$$m_i = \frac{F_i}{\sum_{k=1}^{n} F_k} \tag{6}$$

inwhich n stands for the total number of CB's and $F()$ is the fitness function to be maximized. It may be selected as inverse of a cost function.

Half of the population members are denoted as *Moving CB's* which cane move toward *Stationary* ones (the better/upper CB's after sorting the population in decendeing order of masses). Before collision, stationary CB's have zero velocities.

$$V_i = 0, \quad i = 1, 2, ..., \frac{n}{2} \tag{7}$$

In this stage, velocity of every moving CB is determined by:

$$V_i = X_{i-\frac{n}{2}} - X_i, \quad i = \frac{n}{2} + 1, ..., n \tag{8}$$

After collision, new maximum velocities of colliding bodies are updated due to laws of kinematic energy and momentum conservation by:

$$V_i^{'} = \frac{m_{i+\frac{n}{2}} + \varepsilon m_{i+\frac{n}{2}}}{m_i + m_{i+\frac{n}{2}}} . V_{i+\frac{n}{2}}, \quad i = 1, 2, ..., \frac{n}{2} \tag{9}$$

$$V_i^{'} = \frac{m_i - \varepsilon m_{i-\frac{n}{2}}}{m_i + m_{i-\frac{n}{2}}} . V_{i-\frac{n}{2}}, \quad i = \frac{n}{2} + 1, ..., n \tag{10}$$

inwhich ε stands for the coefficient of resistution defined as the reatio of relative velocity between CB's after collision to such a relative velocity, before collision; e.g.:

$$\varepsilon = \frac{V_i^{'} - V_{i-\frac{n}{2}}^{'}}{V_i - V_{i-\frac{n}{2}}}, \quad i = \frac{n}{2} + 1, ..., n \tag{11}$$

Then position of CB's are updated as:

$$X_i^{new} = X_i + rand.V_i^{'}, \quad i = 1, ..., \frac{n}{2} \tag{12}$$

$$X_i^{new} = X_{i-\frac{n}{2}} + rand.V_i^{'}, \quad i = \frac{n}{2} + 1, ..., n \tag{13}$$

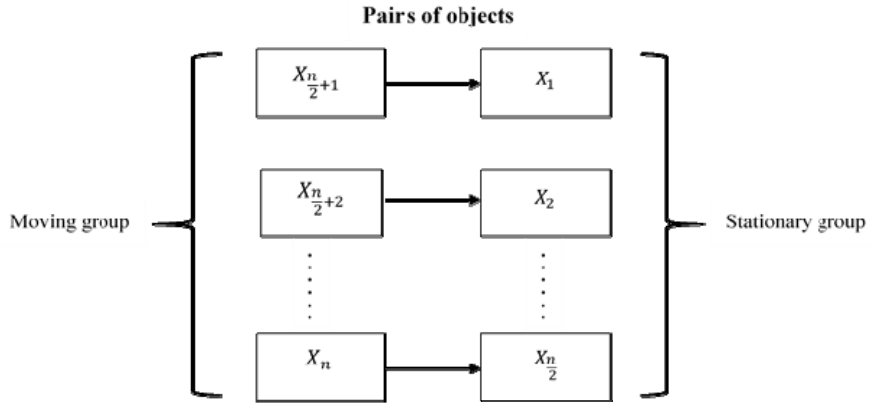

Figure 1. Moving and stationary groups of CB's

Kaveh and Ilchi Ghazaan offered an *Enhanced Colliding Bodies Optimization*, ECBO using an auxiliary memory called *Colliding Memory*, CM[17, 18]. In ECBO, a number of of best CB's during search is saved and replaced by the worst CB's in the current population. Mutation of each design variable by a prescribed probability is also included in ECBO to improve for a better explorative search. For every component of any i^{th} CB vector, If a random number falls below the prescribed probability, P_m, the corresponding j-th design

variable is mutated as:

$$X_{ij} = X_j^{LB} + rand \times (X_j^{UB} - X_j^{LB}) \tag{14}$$

where X^{LB} and X^{UB} are lower and upper bounds on the design vector. In this study, a random number generator such as $rand$ produce continuous values in range [0,1].

3.3 The proposed hybrid colliding bodies optimization and k-means

The present work utilizes a hybrid algorithm to combine global and local search features in the aforementioned algorithms. The method is called ECBO_KM and is introduced via the following steps:

1) Initialization

Generate a randomly distributed population of CB's by the following relation

$$X_i^0 = X^{LB} + rand \times (X^{UB} - X^{LB}), \quad i = 1,...,n \tag{15}$$

Fitness of any i^{th} CB is then evaluated using Eq. (2) and its mass is calculated by Eq. (6). After sorting population of CB's in decsending order of their masses, an auxiliary *Colliding Memory*, CM is initiated that includes the first *CMS* solutions in such a sorted sequence.

2) Main phase

Repeat the following steps for $iter = 1, 2, ..., N_{MaxIter}$, except for the iterations $iter = \beta N_{MaxIter}$ where the enrichment phase is called instead for prescribed $\beta \in \{\beta_1, \beta_2, ..., \beta_L\}$.

2-1) identify moving and stationary CB's in the main population and update their velocities using Eqs. (7)~(10). In this process use the following coefficient of resistution in every iteration *iter*:

$$\varepsilon = 1 - \frac{iter}{N_{MaxIter}} \tag{16}$$

2-2) Update position of CB's using Eqs. (12) ~ (14).

2-3) Evaluate their masses by Eq. (5).

2-4) Generate and sort a temporaily population by adding CM to the current population of CB's. Leave out the *CMS* number of the worst CB's; then update the population and take its best *CMS* number of CB's as the new CM.

3) Enrichment phase

Regarding the best-so-far CB which has been already found in the previous phase centroids of such clusters are calcluated using Eq. (5) and employed as the start of *K-means.*

This algorithm acts as a local search to further improve the design vector for a number of inner iterations. Such an enriched clustering solution is then replaced by the worst CB of the population of the main phase.

4) Convergence check

As soon as the population of colliding bodies is updated via the enrichment phase; control of the algorithm returns back to the main phase. The whole process is repeated until convergence; i.e. reaching a prescribed number of iterations $N_{MaxIter}$.

Note that since each component of the design vector is a cluster ID, it should be rounded to an integer number between e $X^{LB}=1$ and $X^{UB}=K$ in Eqs (12)~(15). Fig. 2 denomstrates flowchart of the proposed hybrid algorithm.

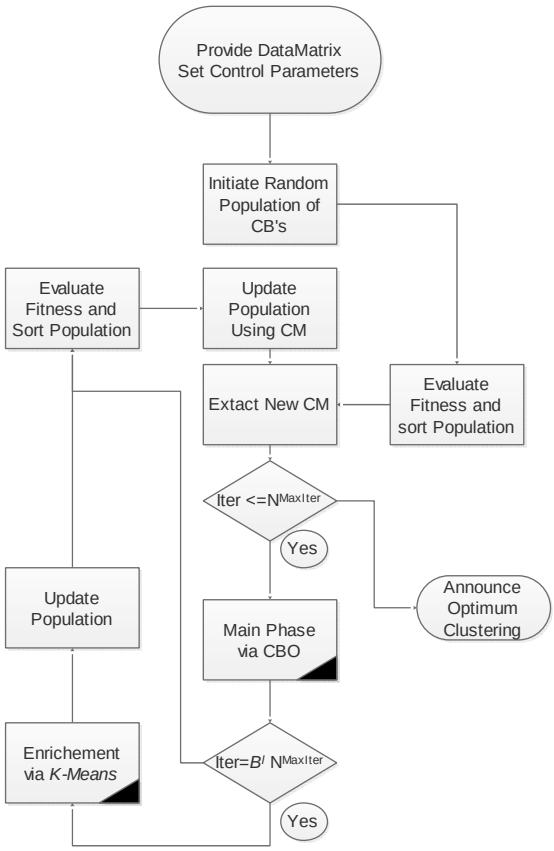

Figure 2. Flowchart of the proposed ECBO-KM

4. NUMERICAL SIMULATION

For strong ground motion clustering a data matrix should be first provided. It includes N_e rows each one corresponding an earthquake in the available database. Henceforth, every

column of such a matrix represents an attribute of the entities (earthquakes).

Consequently, 50 earthquakes are extracted from the PEER catalogue [24] with the magnitude of at least 5 Richter, recorded at less than 20 kilometers from the epicenter. Other attributes include regional features such as soil type, faulting mechanizm, duration of quake, peak ground motion acceleration, displacement and velocity. In addition, *Arias intensity* and *Housner Intensity* are considered as attributes relating to the earthquake input energy. The significant method is applied here for measuring effective duration of earthquakes [25]. The provided datamatrix with such 10 attributes, is further used for clustering of its earthquakes by the three aforementioned algorithms.

Each algorithm is run for 8 different cases of cluster numbers from $K = 4$ to $K = 11$. Table 1 gives the applied control parameters. It reports different population size values, *n*, for different cases of *K*, when CMS is taken about 30% of *n* in each case.

Table 1: Control parameters applied for the optimal clustering problem

n	CMS	P_m	$N_{MaxIter}$	$\{\beta_l\}$
16~24	0.3*n*	0.2~0.3	1500	{0.67}

Table 2 compares the corresponding results of the silhuette sum *Q* and the best fitness obtained by *K*-means, ECBO and ECBO-KM methods. As can be realized from Table 2 and Fig. 4, quality of final solution by *K*-means mostly falls below the ECBO or ECBO-KM. The latter methods have comparable results, however, in most cases ECBO-KM has been the best. The matter is observed specially for larger values of *K*.

Sample behavior of ECBO-KM is compared with ECBO for two cases of $K = 8$ and $K = 9$ in the Fig. 5 and Fig. 6, respectively. It is realized that the proposed ECBO-KM has got capability to escape from premature convergence by applying the enrichment phase at iteration 1000; i.e. for $\beta_1 = \frac{2}{3}$. Note that in the present study, memory enrichment has been employed only once in the ECBO-KM for the sake of more efficiency. By similar reasoning, more quality improvements might be possible if the enrichement is repeated more, however, with the charge of higher total computational effort.

Table 2: Best fitness achieved by different methods in the ground motion clustering

K	*K-means* *Q*	*K-means* *F*	ECBO *Q*	ECBO *F*	ECBO-KM *Q*	ECBO-KM *F*
4	29.6849	0.5937	35.1666	0.7033	36.5704	0.7314
5	30.2925	0.6059	33.8688	0.6774	31.7454	0.6349
6	32.8706	0.6574	33.8688	0.6774	32.5539	0.6511
7	30.4075	0.6082	30.0581	0.6012	32.1155	0.6423
8	27.2517	0.5450	28.3075	0.5661	32.1185	0.6424
9	29.5414	0.5908	32.8150	0.6563	34.3268	0.6865
10	29.3684	0.5874	30.1271	0.6025	34.3456	0.6869
11	30.7413	0.6148	28.8422	0.5768	34.3485	0.6870

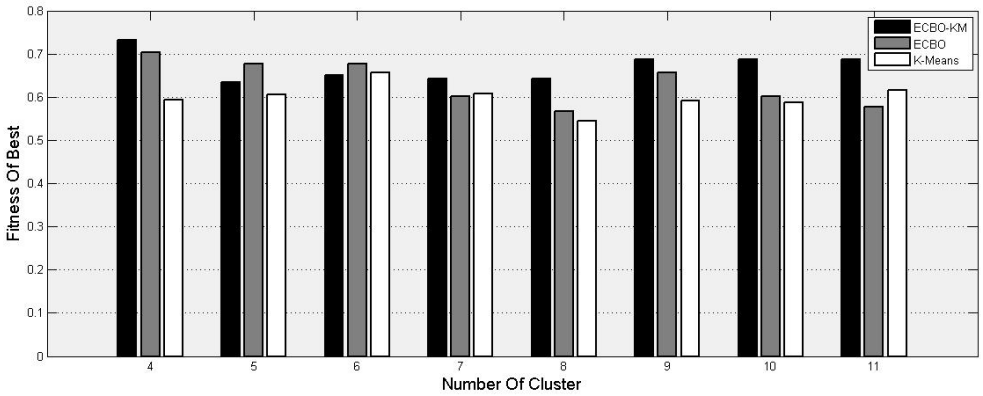

Figure 3. Comparison of the best mean silhouette obtained for different *K* values

In order to compare quality of clustering between the treated algorithms, silhuette plots are provided in Fig. 6 for their best results in the case of $K = 9$. A good clustering contains proper silhuette values; i.e. values closer to +1 and a bad clustering contains negative values near -1. Hence, it is evident from Fig. 6 that the best clustering is resulted by ECBO-KM with positive and near 1 silhuette values. ECBO has some negative *s* values between -0.05 and 0.00 for some of the earthuqakes, but the worst result belongs to the *K*-means which includes some silhuette of -0.20 and even lower.

Differences in clustering may also be observed from amother point of view; that is how uniform is the number of entities in the resulted clusters. In this regard, it can be realized from Fig. 6 that ECBO-KM has been superior to ECBO and K-means. Note that silhuette values for all of the earthquakes in a single cluster are binded close to each other and demonstrated in a sorted manner in the silhuette plot. So the height of every such cluster in this plot is more when it has more members.

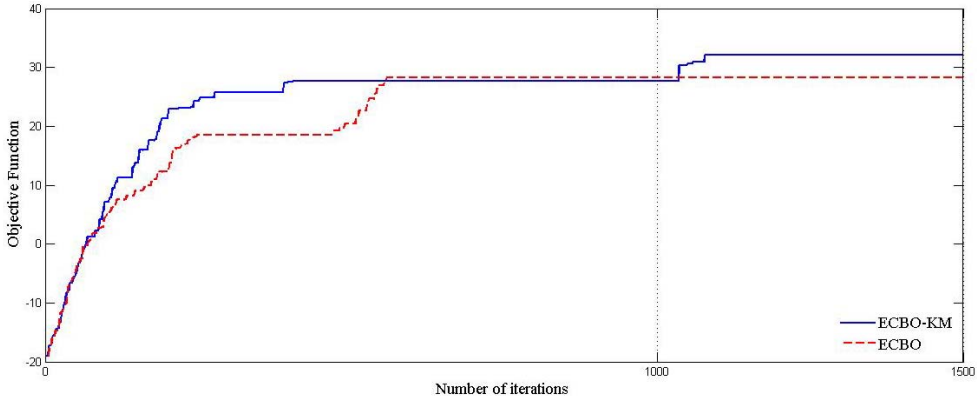

Figure 4. Convergence history of ECBO and ECBO-KM in partitioning the earthquakes into 8 clusters

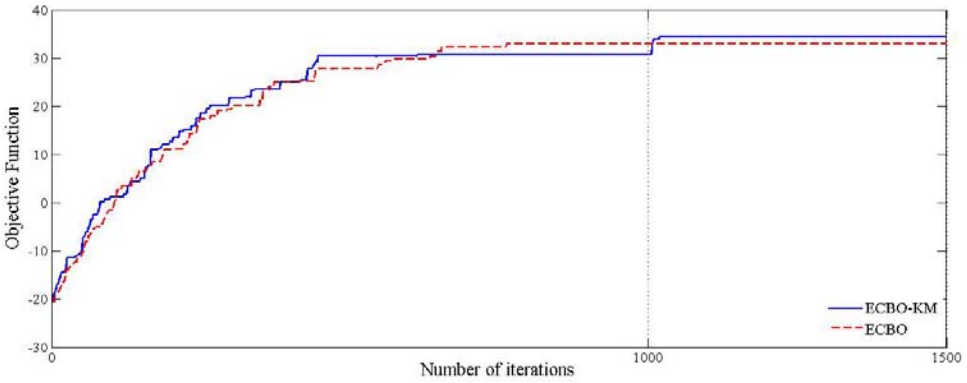

Figure 5. Convergence history of ECBO and ECBO-KM in partitioning the earthquakes into 9 clusters

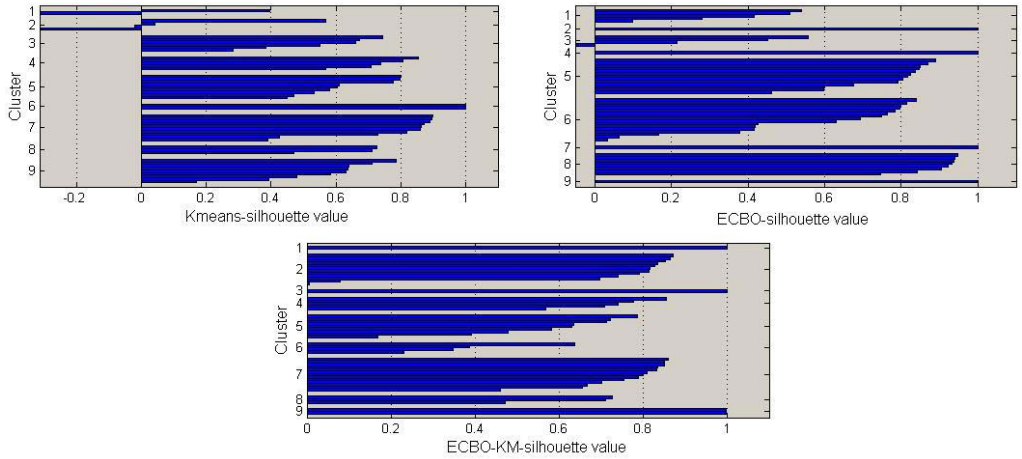

Figure 6. Silhouette plots for *K*-means, ECBO and ECBO-KM in partitioning the earthquakes into 9 clusters

5. CONCLUSION

Due to probabilistic nature of strong ground motions, it will be useful to study them by taking similar observations in different categories. As an unsupervised solution, clustering of earthquakes is concerned in this paper. It is further formulated as an optimization problem using the silhuette sum in the fitness function combined with a utilized penalty to avoid null clusters.

The problem is then treated by three algorithm: *K-means* as a common deterministic clustering algorithm besides to nondeterministic ECBO and ECBO-KM. It is observed that *K*-means can lead to considerably negative silhuette for some of the earthquakes. It acts as a rapid local search method which depends on its starting point. In the other hand, stochastic search in ECBO resulted in better positive silhuette values. However, in some cases of *K*

(number of clusters), *K*-means was better than ECBO.

Henceforth, a hybrid ECBO-KM is proposed in this study to combine suitable features of both K-means and ECBO for the clustering problem; i.e. local search capability of K-means with stochastic search of ECBO. In the ECBO-KM, the best solution via a partial main phase is employed as an enhanced starting point for *K*-means to perform a search refinement. The resulting enriched solution is then substituted in the population of coliding bodies to improve the remainder of such a meta-heuristic search.

As a result, it is observed that ECBO-KM has the capability of escaping from premature convergence toward higher quality solutions; i.e. higher sum of silhuette in its final clustering. In addition, ECBO-KM has resulted in more uniform clusters than *K*-means and ECBO regarding the number of earthquakes in each group. The proposed algorithm can thus be recommended as an enhanced earthquake clustering method which takes benefit of both deterministic refinement and stochastic exploration.

REFERENCES

1. Han J, Kamber M. *Data Mining: Concepts and Tchniques*, Second Edition, Morgan Kaufman Publisher, USA, 2006.
2. Garrett-Mayer E, Parmigiani G. Clustering and classification methods for gene expression data analysis, Working Papers, Department of Biostatics, John Hopkins University, 2004.
3. Jain A, Murty M, Flynn P. Data Clustering: A Review, *ACM Comput Surveys* 1999; **31**(3): 264-323.
4. Jein A, Dubes C. *Algorithms for Clustering Data*, Michigan State University, Prentice Hall, 1988.
5. Ansari A, Noorzad A, Zafarani H. Clustering analysis of the seismic catalog of Iran, *Comput Geosci* 2009; **35**: 475-86.
6. Ansari A, Firuzi E, Etemadsaeed L. Delineation of seismic sources in probabilistic seismic-hazard analysis using fuzzy cluster analysis and monte carlo simulation, *Bull Seismological Soc America* 2015; **105**(4): 2174-91.
7. Kirkpatrick S, Gelatto CD, Vecchi MP. Optimization by simulated annealing, *Sci* 1983; **220**: 671-80.
8. Bandyopadhyay S, Maulik U, Pakhira MK. Clustering using simulated annealing with probabilistic redistribution, *Int J Pattern Recogn Artif Intel* 2001; **15**(2): 269-85.
9. Bandyopadhyay S, Maulik U. Genetic clustering for automatic evolution of clusters and application to image classification, *Pattern Recognition* 2002; **35**(6): 1197-1208.
10. Kennedy J, Eberhart R. *Swarm Intelligence*, Academic Press, London, 2001.
11. Yang XS, *Nature-Inspired Metaheuristic Algorithms*, 2nd Ed, Luniver Press, UK, 2010.
12. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer International Publishing, Switzerland, 2014.
13. Reynolds RG. An Introduction to Cultural Algorithms, *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, World Scienfific Publishing, 1994; pp. 131-139.

14. Kaveh A, Mahdavi VR. *Colliding Bodies Optimization, Extensions and Applications*, Springer International Publishing, Switzerland, 2015.
15. Kaveh A, Mahdavi VR. Colliding bodies optimization method for optimum discrete design of truss structures, *Comput Struct* 2014; **139**: 43-53.
16. Kaveh A, Ilchi Ghazaan M. Enhanced colliding bodies optimization for design problems with continuous and discrete variables, *Adv Eng Softw* 2014; **77**: 66-75.
17. Kaveh A, Ilchi Ghazaan M. Computer codes for colliding bodies optimization and its enhanced version, *Int J Optim Civil Eng* 2014; **4**(3): 321-39.
18. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis University of Fribourg, *J Comput Appl Math* 1987; **20**: 53-65.
19. Hartigan JA, Wong MA. Algorithm AS 136: A K-means clustering algorithm, *J Royal Statistical Society, Series C* 1979; **28**(1): 100-8.
20. Lloyd, SP. Least squares quantization in PCM, *IEEE Trans Info Theory* 1982; **28**(2): 129-37.
21. Hartigan JA. *Clustering Algorithms*, John Wiley & Sons, Inc, 1975.
22. McKay D. *An Example Inference Task: Clustering, Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003: pp. 284-292.
23. Vattani A. k-means requires exponentially many iterations even in the plane, *Discrete Comput Geometry* 2011; **45**(4): 596-616.
24. Pacific Earthquake Engineering Research Center database http://peer.berkeley.edu/smcat/
25. Elnashai AS, Di Sarno L. *Fundamentals of Earthquake Engineering*, John Wiley & Sons Inc, 2008.