The Performance Analysis of a Novel Enhanced Artificial Bee Colony Inspired Global Best Harmony Search Algorithm for Clustering

V. Krishnaveni and G. Arumugam

Department of Computer Science,
Madurai Kamaraj University, Madurai, Tamilnadu, India
{krish.roban,gurusamyarumugam}@gmail.com

Abstract. Clustering is the unsupervised classification of data items of patterns into groups, each of which should be as homogeneous as possible. The problem of clustering has been addressed in many contexts in many disciplines and this reflects its broad appeal and usefulness in exploratory data analysis. This paper presents a new clustering algorithm, called GHSBEEK which is a combination of the Global best Harmony search (GHS) with features of Artificial Bee Colony (ABC) and K-means algorithms. Global-best Harmony search (GHS) is a derivative-free optimization algorithm, which draws inspiration from the musical process of searching for a perfect state of harmony. It has a remarkable advantage of algorithm simplicity. However, it suffers from a slow search speed. The ABC algorithm is applied to improve the members of the Harmony Memory based on their fitness values and hence improves the convergence rate of the Harmony Search method. The GHSBEEK algorithm has been used for data clustering on several benchmark data sets. The clustering performance of the proposed algorithm is compared with the GHS, PSO, and K-means. The simulation results show that the proposed algorithm outperforms the other algorithms in terms of accuracy, robustness, and convergence speed.

1 Introduction

Cluster analysis is a tool for exploring the structure of data. It is a process in which the objects are grouped into clusters such that the objects from the same clusters are similar and objects from different clusters are dissimilar [1]. Clustering is a challenging job in unsupervised learning which is the process of partitioning a set of objects into an apriori unknown number of clusters while minimizing the within-cluster variability and maximizing the between-cluster variability. Clustering has been used in many engineering and scientific disciplines such as Computer Vision, Information Retrieval, Biology and Market Research [1]. Several clustering algorithm categories have been discussed in the literature, including Hierarchical, Partitional, Density-based and Grid-based algorithms [1]. K-means is one of the popular clustering algorithms. But, K-means algorithm is sensitive to the initial states and always converges to the local optimum solution and hence more stochastic search algorithms are being emerged. In this paper, the GHSBEEK algorithm has been

S.C. Satapathy et al. (Eds.): Proceedings of the InConINDIA 2012, AISC 132, pp. 21–28.
springerlink.com

proposed to overcome this problem as well as to solve the clustering problem. Global-best Harmony search (GHS) is a variation of Harmony search (HS) which is a music-based meta-heuristic optimization algorithm. It was inspired by the observation that the aim of music is to search for a perfect state of harmony. This harmony in music is analogous to find the optimality in an optimization process. It has been proved that GHS outperforms HS when applied to high-dimensional problems [8]. This work shows that the diversity maintenance features of ABC can accelerate the convergence speed of the GHS in the proposed method. Further, the performance of Artificial Bee Colony and Harmony search have been analysed and a novel method for clustering in combination with the K-Means algorithm, called GHSBEEK has been proposed.

In sections 2, 3 and 4, Global-best Harmony Search Algorithm, Artificial Bee Colony Algorithm and K-Means Algorithm have been articulated respectively. In section 5, the GHSBEEK algorithm has been proposed and its efficiency and clustering performance have been analysed using bench mark datasets from UCI repository and finally the paper was concluded in section 6.

2 Global-Best Harmony Search

Inspired by the Particle Swarm Optimization, the GHS algorithm was presented with modified pitch adjustment rule. Unlike the basic HS algorithm, the GHS algorithm generates a new harmony vector X_{new} by making use of the best harmony vector in the Harmony Memory (HM) [8].

Main steps of the algorithm are given below:

1: Initialize the problem and algorithm parameters.
2: Initialize the harmony memory.
3: Improvise a new harmony making use of the best harmony vector
4: Update the harmony memory.
5: Repeat steps 3-4 until the stopping criterion is met

3 Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) algorithm [3] for real parameter optimization, is an optimization algorithm which simulates the foraging behaviour of bee colony. The ABC consists of three kinds of bees: employed bees, onlooker bees, and scout bees[3].

In the algorithm, initially, $\mathcal{X}_i = (i = 1, \ldots, SN)$ solutions are randomly produced in the range of parameters where SN is the number of food sources. In the second step of the algorithm, for each employed bee, whose total number equals to the half of the number of food sources, a new source is produced by (1):

$$V_{ij} = x_{ij} + \emptyset_{ij} (x_{ij} - x_{kj}) \tag{1}$$

where $\emptyset_{ij}$ is a uniformly distributed real random number within the range [-1,1] and k is the index of the solution chosen randomly from the colony. After producing V_{ij}, this

new solution is compared to x_{ij} solution and the employed bee exploits the better source. In the third step of the algorithm, an onlooker bee chooses a food source with the probability (2) and produces a new source in selected food source site by (1). For employed bees, the better source is decided by (2):

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \tag{2}$$

where fit_i is the fitness of the solution x_i.

After all onlookers are distributed to the sources, sources are checked whether they are to be abandoned. The employed bee associated with the exhausted source becomes a scout and makes a random search in problem domain by (3).

$$x_{ij} = x_j^{min} + (x_j^{max} - x_j^{min}) * rand \tag{3}$$

4 K-Means Algorithm

The goal of data clustering is grouping data into a number of clusters and *K*-means algorithm is the most popular clustering algorithm. Let $X = (x_1, x_2, \ldots\ldots, x_N)$ be a set of N data and let each data vector be a *p*-dimensional vector. Let $C = \{c_1, c_2, \ldots, c_k\}$ be a set of *K* clusters and *K* denotes the number of cluster centroids which is provided by the user. In *K*-means algorithm, *K* cluster centroid vectors are initialized randomly and then each data vector to the class is assigned with the closest centroid vector [8]. In this study, Euclidian metric has been used as a distance metric. The expression is given in (4)

$$D(x_i, c_j) = \sqrt{\sum_{k=1}^{P} (x_{ik} - c_{jk})^2} \tag{4}$$

After all data are being grouped, the cluster centroid vectors are recalculated using (5)

$$\mathbf{C_j} = \frac{\mathbf{1}}{\boldsymbol{n_j}} \sum_{\forall \boldsymbol{x_{i \in xc_j}}} \boldsymbol{x_i} \tag{5}$$

where n_j is the number of data vectors which belong to cluster *j*. After the above process, the data to the new cluster centroids are reassigned and the process is repeated until a criterion is satisfied. To measure the goodness of the partition, a measure must be defined. A popular performance function for measuring goodness of the partition is the total within-Cluster variance or the total mean-square quantization error (MSE), which is defined in (6).

$$\text{Perf}(X,C) = \sum_{i=1}^{N} \boldsymbol{Min}\{\|\boldsymbol{X}_i - \boldsymbol{C}_l\|^2 \mid l = 1, \ldots\ldots, K\} \tag{6}$$

5 The Proposed Algorithm - GHSBEEK

5.1 The Idea Behind

In GHS, the update of the Harmony memory highly depends on the past search experiences. Unfortunately, this inherent shortcoming limits the search ability of the GHS method. The food source exploitation feature of the Artificial Bee Colony method is employed to improve the fitness of the solution candidates in the HM. While the ABC inspired GHS algorithm can be used as a global search strategy across the whole solution space, the K-Means algorithm has been used as a local strategy for improving solutions. The following Pseudo code illustrates the GHSBEEK

(i) Initialize the Harmony Memory (HM) with initial centroids selected randomly from the original data set
Execute K-Means and calculate fitness for each solution vector
(ii) Improvise a new Harmony: Define centroids for this solution
for I = 1 to D do (where D represents Dimension)
if (rand >HMCR)
begin
Randomly select a vector solution from HM
Use the food source exploitation feature of ABC mutate the vector by its neighbouring centroid values within limits
Execute K-Means and calculate fitness of the mutated solution
Compare the fitness values of mutated vector and the randomly selected one
Newcentroid [I] =mutated vector if it has better fitness value else the randomly selected one
if (rand>PAR)
Generate a Newcentriod[I] using the best harmony vector
endif
end
else
Newcentroid[I] = Randomly selected vector solution from HM
endif
Next-for
Execute K-Means and Calculate fitness for new harmony
(iii) Update the harmony memory
(iv) Check the Stopping Criterion: If the maximum number of improvisations is satisfied, Iteration is terminated else repeat steps (ii) and (iii)
(v) Select the best Harmony in HM: find the best harmony
Execute K-Means and Calculate fitness for best harmony.
(vi) Return the best harmony in harmony memory

5.2 Data Clustering and Experimental Setup

The Proposed algorithm has been implemented using MATLAB 7.0 and three data sets were selected from the UCI machine learning repository [12].

For GHS algorithm, parameters were set to the values recommended in [8]. Size of the harmony memory was 15, HMCR = 0.9, PAR = 0.3, BW = 0.01 and the maximum improvisation number was 10000 for all test problems. For the proposed algorithm, the same parameter setting has been maintained. The standard PSO has been used. In this algorithm, the inertia weight ω varies from 0.9 to 0.7 linearly with the iterations and the acceleration factors $c1$ and $c2$ have been kept as 2.0 [6].

The performance evaluation of the proposed GHSBEEK approach for clustering on three different data sets was done and its results were compared with the results of the K-means, PSO, and GHS clustering algorithms.

Motorcycle data (N = 133, d = 2, K = 4): the Motorcycle benchmark consists of a sequence of accelerometer readings through time following a simulated motorcycle crash during an experiment to determine the efficacy of crash helmets.

Iris data (N =150, d =4, K = 3): this data set is with 150 random samples of flowers from the iris species setosa, versicolor, and virginica. From each species there are 50 observations for sepal length, sepal width, petal length, and petal width in cm.

Wine data (N =178, d =13, K =3): There are 178 instances with 13 numeric attributes in wine data set. All attributes are continuous. There is no missing attribute value.

For every data set, each algorithm has been applied 30 times individually with random initial solution. Table 1 summarizes the intracluster distances, as defined in (6), obtained from all algorithms for the data sets above. The average, best, and worst solution of fitness from 30 simulations, and standard deviation have been presented in Table 1. Fig. 1, 2 and 3 show the search progress of the average values found by four algorithms over 30 runs for three data sets.

5.3 Experimental Results

From the values in Table 1, it has been concluded that the results obtained by GHSBEEK are clearly better than the other algorithms for all data sets; GHS is a little better than PSO; the K-means is the worst for all data sets.

For MotorCycle data set, the optimum of the fitness function for all algorithms, except K-means, is 2.060e+003. From the values of the standard deviation, it is observed that the GHSBEEK algorithm is performing better than the other methods. The standard deviation value of GHSBEEK, which is less than 1 represents that the algorithm is converged to the global optimum most of the times.

For Iris data set, GHSBEEK and GHS provide the optimum values and small standard deviation when compared to those of obtained by other methods. The average values of the fitness function for GHSBEEK and GHS are 0.927e+002 and 0.930e+002 respectively; the standard deviations for GHSBEEK and HS algorithms are less than 1 which indicates that GHSBEEK and GHS are converged to the global optimum most of the times.

For Wine data set, the results of GHSBEEK algorithm have outperformed the other methods. It has converged to Global optimum most of the times compared to other methods.

Finally, from the graphs shown in Fig. 1, 2 and 3 for all data sets, it has been concluded that GHSBEEK outperforms the other three methods as it converges to the optimal value in a faster manner

Table 1. Comparison of intracluster distances for the four clustering algorithms

Data set	Criteria	GHSBEEK	GHS	PSO	K-means
Motor Cycle	Average	2.078e + 003	2.854e + 003	2.976e + 003	3.412e + 004
	Best	2.070e + 003	2.070e + 003	2.077e + 003	3.187e + 004
	Worst	2.224e + 003	2.934e + 003	3.053e + 003	3.658e + 004
	Std	1.176e + 001	1.198e + 001	1.549e + 001	2.623e + 001
Iris	Average	0.927e + 002	0.930e + 002	0.975e + 002	1.342e + 002
	Best	0.904e + 002	0.904e + 002	0.921e + 002	1.067e + 002
	Worst	0.935e + 002	0.947e + 002	1.053e + 002	1.725e + 002
	Std	1.942e - 001	1.754e + 000	1.7629 + 000	1.736e + 001
Wine	Average	1.652e + 003	1.673e + 003	1.342e + 004	1.642e + 004
	Best	1.603e + 003	1.606e + 003	1.297e + 004	1.607e + 004
	Worst	1.697e + 003	1.698e + 003	1.363e + 004	1.684e + 004
	Std	1.917e - 002	1.146e + 000	1.128e + 001	1.926e + 001

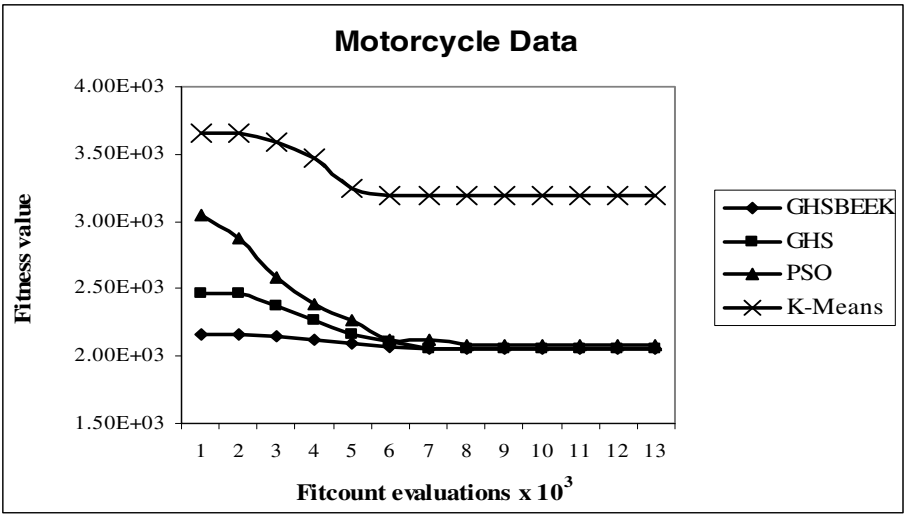

Fig. 1. Comparing the convergence of the proposed GHSBEEK based clustering with other approaches in terms of total Mean-Square quantization Error for Motorcycle data set

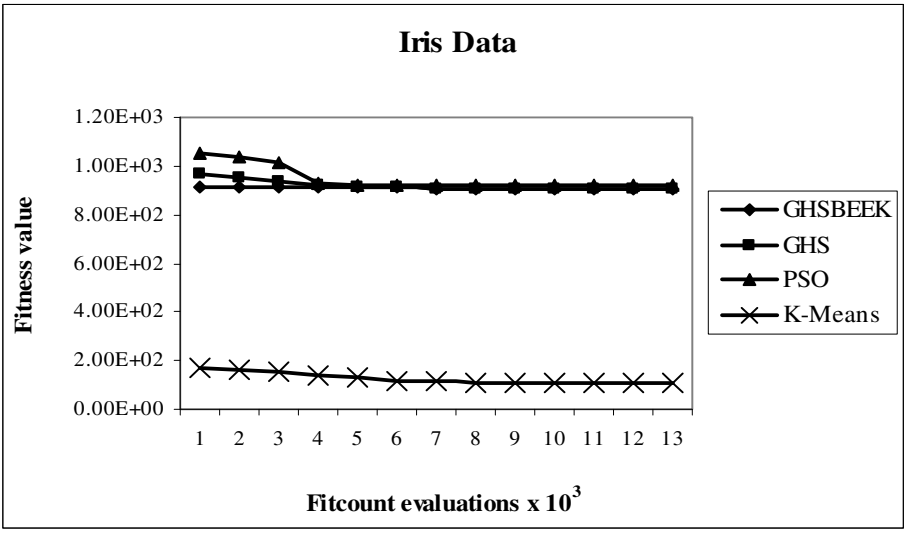

Fig. 2. Comparing the convergence of the proposed GHSBEEK based clustering with other approaches in terms of total Mean-Square quantization Error for Iris data set

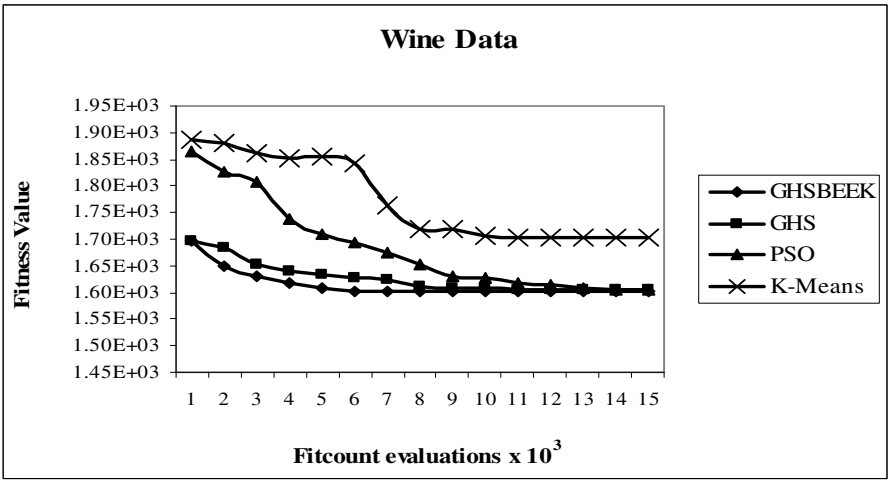

Fig. 3. Comparing the convergence of the proposed GHSBEEK based clustering with other approaches in terms of total Mean-Square quantization Error for Wine data set

6 Conclusion and Future Work

This paper presented a novel algorithm GHSBEEK for solving data clustering problem. The performance of GHS algorithm has been increased by employing the food source exploitation feature of the ABC algorithm which improves the members of the Harmony Memory based on their fitness values and hence improves the

convergence rate of the Global-best Harmony Search method. The exploitation process has been carried out in a controlled way so that the better harmony vectors enjoy the higher selection probability. These actions enabled the speedy update of harmony memory with better solutions and hence caused the search process to move rapidly towards the goal. This enhancement also avoids the problem of getting trapped into the local minima, as the chance of selecting the same harmony vector repeatedly has been minimized. This results in an optimization algorithm which can be used for solving multivariable, multimodal function optimization. This algorithm, in combination with the K-Means clustering algorithm showed significant improvements in the performance in terms of solution quality and convergence speed compared to other optimization algorithms in the data clustering process.

There are many tasks for future work; The GHSBEEK can be applied to real data sets; a metric can be included so that the number of clusters can be found automatically; the other variants of HS such as SGHS and IHS can be used instead of GHS; K-Medoids or Expectation Maximization algorithms can be used instead of K-Means and the results can be compared; finally, the concept of Feature Selection can be included.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM, 0360-0300/99/0900-0001 (2000)
2. Bratton, D., Kennedy, J.: Defining a Standard for Particle Swarm Optimization. In: Proc. Of the IEEE Swarm Intelligence Symposium (SIS), pp. 120–127 (2007)
3. Karaboga, D., Basturk, B.: Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 789–798. Springer, Heidelberg (2007)
4. Karaboga, D.: An idea based on honey bee swarm for Numerical optimization. Technical Report TR06, Erciyes University, Engineering faculty, Computer Engineering Department (2005)
5. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering 194(36-38), 3902–3922 (2005)
6. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
7. Liu, K., Tan, Y., He, X.: Particle Swarm Optimization Based Learning Method for Process Neural Networks. In: Zhang, L., Lu, B.-L., Kwok, J. (eds.) ISNN 2010, Part I. LNCS, vol. 6063, pp. 280–287. Springer, Heidelberg (2010)
8. Omran, M.G.H., Mahdavi, M.: Global-best Harmony Search. Appl. Math. Comput. 198, 643–656 (2008)
9. Redmondand, S.J., Heneghan, C.: A method for initializing the K-means clustering algorithm using kd trees. Pattern Recognition Letters 28, 965–973 (2007)
10. Kang, S.L., Geem, Z.W.: A new structural optimization method based on the Harmony search Algorithm. Computers and Structures 82(9-10), 781–798 (2004)
11. Geem, Z.W., Kim, J.H., Loganathan, G.V.: Harmony Search Optimization: application to pipe network design. International Journal of Modeling and Simulation 22(2), 125–133 (2002)
12. UCI Machine Learning Repository: datasets, `http://archive.ics.uci.edu/ml/datasets.html`