

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)**SciVerse ScienceDirect**

Procedia Technology 6 (2012) 265 – 274

**Procedia**  
Technology**2nd International Conference on Communication, Computing & Security [ICCCS-2012]**

## Effect of Harmony Search Parameters' Variation in Clustering

Vijay Kumar<sup>a</sup>, Jitender Kumar Chhabra<sup>b</sup>, Dinesh Kumar<sup>c</sup>: a\*<sup>a</sup> Computer Science & Engineering Department, JCDM College of Engineering, Sirsa, Haryana, India<sup>b</sup> Computer Engineering Department, National Institute of Technology, Kurukshetra, Haryana, India<sup>c</sup> Computer Science & Engineering Department, GJUS&T, Hisar, Haryana, India

### Abstract

Harmony search (HS) is a relatively new meta-heuristic optimization method, which is based on the concept of music improvisation. This paper depicts the impact of constant parameters such as Harmony Memory Consideration Rate and Pitch Adjusting Rate, and presents an approach for parameter tuning. It presents modifications in existing harmony search, by choosing appropriate values of these two parameters and allows them to change dynamically during the process of improvisation. The proposed algorithm has been evaluated for data clustering on five benchmark datasets. The clustering performance of proposed algorithm is compared with K-Means, Genetic algorithm, HS and improved version of HS. Experimental results reveal that proposed algorithm provides better results than the above said techniques in terms of precision, recall, G-Measure, inter-cluster and intra-cluster distance.

© 2012 The Authors. Published by Elsevier Ltd. Selection and/or peer-review under responsibility of the Department of Computer Science & Engineering, National Institute of Technology Rourkela. Open access under [CC BY-NC-ND license](http://creativecommons.org/licenses/by-nc-nd/4.0/).

*Keywords:* Meta-heuristics; Clustering; Harmony Search

### 1. Introduction

Meta-heuristic algorithms are search methods simulating from nature and are becoming more popular in recent years. The major components of meta-heuristic algorithms are Intensification and diversification (Yang, 2010). Intensification intends to search around the current best solutions and selects the best solution. Diversification avoids solution being trapped in the local optima and increases the diversity of solutions. The best combination of these ensures that the global optimality is achievable. The Meta-heuristic algorithms such

\* Corresponding author. Tel.: +91-9466255106

E-mail address: [vijaykumarchahar@gmail.com](mailto:vijaykumarchahar@gmail.com), [jitenderchhabra@gmail.com](mailto:jitenderchhabra@gmail.com), [dinesh\\_chutani@yahoo.com](mailto:dinesh_chutani@yahoo.com).

as Genetic algorithm, Differential Evolution, and Particle Swarm Optimization have been applied to a wide variety of optimization problems (Osman and Laporte, 1996; Blum and Andrea, 2003; Yagiura and Ibaraki, 2001).

Harmony Search (HS) is a relatively new meta-heuristic algorithm. It is inspired from music improvisation process where the musicians improvise their instruments' pitches to attain perfect state of harmony (Omran and Mahdavi, 2008). It possesses several advantages over the traditional optimization techniques including (Geem et al., 2001; Mahdavi et al., 2007; Malaki et al., 2008) such as (1) it is a simple meta-heuristic algorithm and does not require for initial setting for decision variables, (2) it uses stochastic random searches, so derivative information is unnecessary, (3) it has few parameters for fine-tuning. Due to these features, performance of HS has been found to be better than earlier existing meta-heuristic algorithms. HS algorithm was further improved by Mahdavi et al., 2007 in form of Improved Harmony Search (IHS) by dynamically changing parameters. However, in IHS the effect of harmony memory consideration rate (HMCR) is not considered. HMCR is a major and dominating factor in the optimization process by harmony search (Amiri et al., 2010).

This paper proposes a modified version of IHS that exploits variable pitch adjustment rate and harmony memory consideration rate. Thereafter modified version of IHS (MHS) is applied on clustering problem. The MHS is compared with K-Means, Genetic algorithm based clustering and the versions of HS proposed by Geem et al., 2001 and Mahdavi et al., 2007. The remainder of this paper is organized as follows. Section 2 presents the clustering problem. Section 3 provides an overview of HS and its variant. Section 4 presents the proposed approach and its time complexity. The application of MHS on clustering problem is also discussed in section 4. Section 5 covers the experimental results followed by conclusions in section 6.

## 2. Clustering Problem

Clustering is a distribution of data into groups based upon similar characteristics of data while minimizing the variability within group, called intra-cluster distance and maximizing the between group variability, called inter-cluster distance. Let the set of  $n$  input data points  $X = \{x_1, x_2, \dots, x_n\}$ , where  $x_j = (x_{j1}, x_{j2}, \dots, x_{jd}) \in R^d$ , with each measure  $x_{ji}$  called a feature. Clustering algorithm tries to find out a  $K$ -partition of  $X$ ,  $C = \{C_1, C_2, \dots, C_K\}$  such that (Jain et al., 1999; Xu and Wunsch, 2009):

$$C_i \cap C_j = \phi, \quad i, j = 1, 2, \dots, K \text{ and } i \neq j$$

$$\bigcup_{i=1}^K C_i = X$$

Most of the clustering algorithms are sensitive toward initially randomly selected cluster centers. Search ability of heuristic algorithms is used to solve this problem. In this paper, euclidean metric is used as distance measure and squared error function as an objective function. The clustering algorithm aims to minimize the objective function (Amiri et al, 2010).

$$f(X, C) = \sum_{i=1}^n \min \left\{ \|x_i - c_j\|^2 \mid j = 1, \dots, K \right\} \quad (1)$$

where  $\|x_i - c_j\|^2$  is a distance measure between cluster center  $c_j$  and data point  $x_i$ .

### 3. Harmony Search and its Proposed Modified Version

This section first describes the brief overview of HS and its variant followed by the proposed modification in HS.

#### 3.1. Harmony Search Algorithm and its Variant

The concept of Harmony Search Algorithm (HS) was first presented by Geem et al., 2001. It is a meta-heuristic algorithm inspired by the improvisation process of music players. In HS, each decision variable generates a value for finding global optimum. It has been applied to many optimization problems such as traveling salesman problem, chaotic systems, economic power dispatch, vehicle routing and sudoku problem. The HS algorithm consists of following steps (Geem et al., 2001; Mahdavi et al., 2007):

- Step 1: **Initialize the problem and algorithm parameters:** The optimization problem is defined as minimize (or maximize)  $f(x)$  such that  $LB_i \leq x_i \leq UB_i$ , where  $f(x)$  is the objective function,  $x$  is a candidate solution consisting of  $N$  decision variables  $x_i$ .  $LB_i$  and  $UB_i$  are the lower and upper bounds for each decision variables respectively. In addition, the parameters of the HS are Harmony Memory Size (HMS), Harmony Memory Considering Rate (HMCR), Pitch Adjusting Rate (PAR), distance Bandwidth (BW) and the Number of Improvisations (NI) or stopping criterion.
- Step 2: **Initialize the Harmony Memory:** The Harmony Memory (HM) is a memory location, where the solution vectors are stored. HM is similar to the genetic pool in GA (Geem et al., 2001). The initial HS is generated from uniform distribution in ranges, where. This is achieved as follows:

$$x_i^l = LB_i + Rand \times (UB_i - LB_i), \quad l = 1, 2, \dots, HMS \quad (2)$$

- Step 3: **Improve a New Harmony:** Generating a new harmony is called improvisation. The new harmony vector,  $x' = (x'_1, x'_2, \dots, x'_N)$ , is generated based on three rules: memory consideration, pitch adjustment and random selection. The procedure works as follows:

```

For each  $i \in [1, N]$  do
  if  $U(0,1) \leq HMCR$  then
    begin
       $x'_i = x_i^l$ , where  $l \sim U(1, \dots, HMS)$ 
    end
  if  $U(0,1) \leq PAR$  then
    begin
       $x'_i = x_i \pm Rand \times BW$ 
    end
  else
     $x'_i = LB_i + Rand \times (UB_i - LB_i)$ 
  endif
done

```

- Step 4: **Update the Harmony Memory:** The new harmony vector replaces the worst harmony vector in the HM, only if its fitness is better than the second one.
- Step 5: **Check the Stopping Criterion:** Terminate when maximum number of improvisations is reached. Otherwise, steps 3 and 4 are repeated.

To improve the performance of the HS, Mahdavi et al. proposed a new variant of HS, called Improved Harmony Search (IHS). The IHS dynamically updates PAR and BW in improvisation step according to following equations (Mahdavi et al., 2007):

$$PAR(gn) = PAR_{\min} + \frac{(PAR_{\max} - PAR_{\min})}{NI} \times gn \quad (3)$$

where  $PAR(gn)$  is the pitch adjusting rate for generation  $gn$ ,  $PAR_{\min}$  is the minimum adjusting rate,  $PAR_{\max}$  is the maximum adjusting rate,  $NI$  indicate for the number of improvisation and  $gn$  represents current generation/iteration number.

and

$$BW(gn) = BW_{\max} e^{\left( \frac{\ln\left(\frac{BW_{\min}}{BW_{\max}}\right)}{NI} \times gn \right)} \quad (4)$$

where  $BW(gn)$  is the bandwidth for generation  $gn$ ,  $BW_{\min}$  is the minimum bandwidth,  $BW_{\max}$  is the maximum bandwidth.

A major drawback of the IHS is that HMCR is fixed. To improve the performance of IHS, modifications are proposed in preceding subsection.

#### 4. Modified Harmony Search Algorithm

Performance of Harmony Search heavily depends on PAR and HMCR. The HMCR and PAR parameters are used for global searching and improving local solutions respectively (Geem et al., 2001). Therefore, fine-tuning of these parameters can be quite useful. This paper proposes a modification in the improvisation step of the Harmony Search. The proposed modification is based on the method of adjustment of PAR and HMCR values.

The adjustments in parameters are based on the concept of meta-heuristic approaches. According to this, first algorithm has to explore the entire search space in earlier iteration. After some iteration, it must confine to some local space. In earlier generations, HMCR must be small so that each solution must be explored whether it is present in HM or not. Small HMCR means the larger probability of choosing the values from the entire feasible range. The global best solutions, which were not initially stored in HM, are stored in HM as algorithm progresses. The number of solutions stored in HM is changed with generations. The value of HMCR should be increased with the increase in number of generations so that search confined to local memory (HM). Towards the final generations, larger HMCR values indicate the larger probability of choosing the values from within HM only as the possible best solutions, found during the earlier generations, are in the HM. Our proposed formula ensures the same in a way that initially the values are taken from entire feasible range, not confined to

HM only. If these values are better than those already existing in HM, are replaced with existing worst entries for the obvious reasons.

PAR is performed on values chosen from the HM and (1-PAR) sets the rate of doing nothing (Geem et al., 2001). Higher value of PAR means that pitch adjustment is required for each values obtained above. Small PAR values mean that the decision values chosen from HM using HMCR don't require modification. In earlier generations, PAR must be high as the solutions, which are chosen from feasible range and not from HM, require more pitch adjustment operations. As number of generation increases, best solutions keep on adding and the HM contains a large number of best solutions. This results in reduction in pitch adjustment operation requirement. Furthermore, larger HMCR values with small PAR values leads to further improvement in best solutions in final generations and these best solutions are converged to optimal solutions.

To improve the performance of IHS algorithm, modification is proposed in the form of variations in the values of PAR and HMCR during improvisation step. Small PAR and large HMCR provide optimal solution in final iteration. Values of PAR and HMCR are lies between the maximum and minimum range specified for them. PAR and HMCR change dynamically with generation number as shown in figure 1 and expression is given below:

$$PAR(gn) = PAR_{\min} + \frac{(PAR_{\max} - PAR_{\min})}{NI} \times (NI - gn) \quad (5)$$

and

$$HMCR(gn) = HMCR_{\min} + \frac{(HMCR_{\max} - HMCR_{\min})}{NI} \times gn \quad (6)$$

where  $HMCR(gn)$  is the harmony memory consideration rate for generation  $gn$ ,  $HMCR_{\min}$  is the minimum harmony memory consideration rate,  $HMCR_{\max}$  is the maximum harmony memory consideration rate.

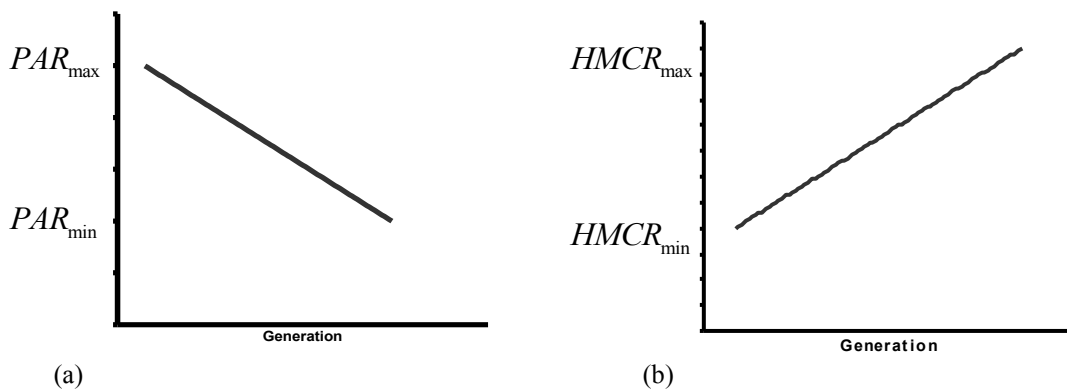


Fig. 1. (a) PAR versus Generation; (b) HMCR versus Generation

#### 4.1. Algorithm

The optimization procedure of the Modified Harmony Search (MHS) algorithm consists of steps as follows:

- Step 1: Initialize the optimization problem and algorithm parameters.
- Step 2: Initialize the harmony memory (HM).
- Step 3: Improvise a new harmony from HM using equations 5 and 6.
- Step 4: Update the HM.
- Step 5: Repeat Steps 3 and 4 until the termination criterion is satisfied.

#### 4.2. Time Complexity

The major computational load during generation of the proposed MHS is in the improvisation of new harmony procedure and fitness evaluation. The improvisation of new harmony and fitness evaluation of a given solution takes  $O(nd)$  and  $\theta(nd)$  time, respectively. Where  $n$  is number of data points, and  $d$  is dimensions of given problem. Therefore, the overall complexity of proposed MHS is  $\theta(ndpg)$ , where  $p$  is population size, and  $g$  is the number of generations.

#### 4.3. Application of Modified Harmony Search on Clustering

The harmony search algorithm is used for finding optimal cluster centers when the number of cluster to be known in advance. In this paper, we have considered each cluster center as a decision variable. Therefore, each row of HM contains  $K$  decision variables that represent one possible solution. A row of matrix,  $C = (C_1, \dots, C_K) = (c_{11}, \dots, c_{1d}, c_{21}, \dots, c_{2d}, \dots, c_{k1}, \dots, c_{kd})$ , number of decision variables is  $K \times d$ . where  $K$  is the number of clusters and  $d$  is dimension of the dataset. The performance of clustering is measured by objective function that is mentioned in equation 1. The HM is filled with random chosen data points from given dataset. A new harmony vector  $c_{new} = (c_{11}^n, \dots, c_{1d}^n, c_{21}^n, \dots, c_{2d}^n, \dots, c_{k1}^n, \dots, c_{kd}^n)$  is generated using improvisation process. If the new harmony vector is better than harmony in HM in term of objective function, the new harmony is included otherwise excluded from HM. Computation is terminated when the maximum number of improvisation is satisfied.

### 5. Experimentation and Results

This section compares the performance of modified harmony search based clustering (MHSC) with K-Means, harmony search based clustering (HSC), improved harmony search based clustering (IHSC), and Genetic algorithm based clustering (GAC). The results are evaluated and compared using some widely acceptable cluster quality measure such as precision, recall, weighted average and G-Measure (Buckland and Gay, 1994; Kowalski, 1997). Large value of these measures is required for better clustering. To prove the efficacy of MHSC, it is also compared with K-Means, GAC, HSC and IHSC in term of inter-cluster and intra-cluster distance. Smaller value of intra-cluster and large value of inter-cluster distance is required for better clustering.

### 5.1. Real Life Datasets

All the clustering techniques used in this paper have been tested over five real-life datasets of UCI database (Blake and Merz, 1998). Table 1 presents the details of these datasets.

Table 1. UCI Machine Learning Datasets

Dataset	Number of Instances	Number of Features	Number of Classes
Wine	178	13	3
Haberman	306	3	2
Bupa Liver disorders	345	6	2
CMC	1473	9	3
Breast Cancer	683	9	2

### 5.2. Parameter Setting

For the HSC, HMCR is taken as 0.9, PAR as 0.5 and BW as 0.01, as recommended in (Mahdavi et al., 2007). For the MHSC and IHSC, PAR is assigned range [0.01, 0.99], recommended in (Mahdavi et al., 2007). The maximum number of iteration and size of HM for HSC, IHSC and MHSC are fixed as 100 and 15 respectively. The value of BW for IHSC and MHSC is set to [0.001, 0.1] as these are commonly used by the most of the researchers in the recent time. For the MHSC, HMCR is assigned a range [0.5, 0.95] based on our experimentation. For the GAC, population size is set to 15. The probability of crossover and mutation is set to 0.8 and 0.01 respectively, which is reported in literature.

### 5.3. Results and Discussion

Tables 2-6 show the comparison between proposed MHSC approach and above said techniques in terms of cluster quality measures for wine, haberman, bupa, CMC, and breast cancer datasets respectively. The results have been compared in terms of means and standard deviation over 10 independent runs in each case. For wine, CMC and breast cancer dataset, it has found that the MHSC provides much better cluster quality than the other clustering algorithms. GAC and MHSC perform comparable results for haberman dataset. For bupa dataset, it is found that weighted average is good for GAC, but with higher variance than the MHSC.

Table 2. Mean and standard deviation of cluster quality matrices for Wine Dataset

	Precision	Recall	Weighted Average	G-Measure
K-Means	0.27180 (0.13649)	0.27418 (0.14095)	0.26910 (0.13764)	0.07770 (0.2457)
GAC	0.29249 (0.21913)	0.31052 (0.23234)	0.30727 (0.22719)	0.10820 (0.24451)
HSC	0.35688 (0.25536)	0.35787 (0.23122)	0.36292 (0.23365)	0.21108 (0.31531)
IHSC	0.38791 (0.21943)	0.39235 (0.19050)	0.39101 (0.19528)	0.21887 (0.28360)
MHSC	<b>0.43425</b> <b>(0.21084)</b>	<b>0.40868</b> <b>(0.18966)</b>	<b>0.41471</b> <b>(0.19518)</b>	<b>0.25192</b> <b>(0.29401)</b>

Table 3. Mean and standard deviation of cluster quality matrices for Haberman Dataset

	Precision	Recall	Weighted Average	G-Measure
K-Means	0.49722 (0.01409)	0.49642 (0.01797)	0.48366 (0.00846)	0.45568 (0.01611)
GAC	0.49948 (0.01806)	0.49938 (0.02299)	0.49958 (0.03969)	0.46466 (0.02255)
HSC	0.49550 (0.02260)	0.49534 (0.02788)	0.50884 (0.05008)	0.45963 (0.03403)
IHSC	0.49439 (0.01297)	0.49318 (0.01630)	0.51961 (0.03216)	0.45768 (0.01838)
MHSC	<b>0.49961</b> <b>(0.01575)</b>	<b>0.49940</b> <b>(0.01899)</b>	<b>0.50785</b> <b>(0.02595)</b>	<b>0.46027</b> <b>(0.02087)</b>

Table 4. Mean and standard deviation of cluster quality matrices for Bupa Dataset

	Precision	Recall	Weighted Average	G-Measure
K-Means	0.50000 (0.01317)	0.50000 (0.00822)	0.50000 (0.04427)	0.43020 (0.01191)
GAC	0.47874 (0.04630)	0.48916 (0.02708)	0.50888 (0.03729)	0.38385 (0.05137)
HSC	0.50783 (0.04900)	0.50146 (0.02660)	0.47566 (0.03161)	0.41080 (0.03624)
IHSC	0.48798 (0.04812)	0.49171 (0.02279)	0.49815 (0.04191)	0.38449 (0.03029)
MHSC	<b>0.51197</b> <b>(0.04655)</b>	<b>0.50541</b> <b>(0.01953)</b>	<b>0.50232</b> <b>(0.030764)</b>	<b>0.43237</b> <b>(0.04914)</b>

Table 5. Mean and standard deviation of cluster quality matrices for CMC Dataset

	Precision	Recall	Weighted Average	G-Measure
K-Means	0.34323 (0.04172)	0.33203 (0.04438)	0.34704 (0.04632)	0.31392 (0.05692)
GAC	0.33782 (0.05995)	0.33724 (0.05700)	0.32939 (0.05595)	0.31958 (0.05883)
HSC	0.35599 (0.01367)	0.35202 (0.01442)	0.36714 (0.006387)	0.33524 (0.02180)
IHSC	0.33786 (0.07007)	0.34174 (0.06549)	0.34767 (0.06402)	0.31834 (0.07191)
MHSC	<b>0.36154</b> <b>(0.01852)</b>	<b>0.35694</b> <b>(0.018370)</b>	<b>0.36728</b> <b>(0.00898)</b>	<b>0.34585</b> <b>(0.02228)</b>

Table 6. Mean and standard deviation of cluster quality matrices for Breast Cancer Dataset



	Precision	Recall	Weighted Average	G-Measure
K-Means	0.31554 (0.44589)	0.31870 (0.43861)	0.31552 (0.44604)	0.31425 (0.44409)
GAC	0.41006 (0.46431)	0.41673 (0.43411)	0.41244 (0.45465)	0.39825 (0.46052)
HSC	0.40859 (0.47144)	0.41204 (0.45491)	0.40938 (0.46801)	0.40417 (0.46795)
IHSC	0.40711 (0.47289)	0.41141 (0.45786)	0.40850 (0.47005)	0.40320 (0.47023)
MHSC	<b>0.49924</b> <b>(0.48157)</b>	<b>0.50011</b> <b>(0.46392)</b>	<b>0.49870</b> <b>(0.47761)</b>	<b>0.49380</b> <b>(0.47832)</b>

Tables 7-11 show the comparison between MHSC with other clustering techniques for above-mentioned five benchmark datasets in terms of intra and inter-cluster distance. MHSC provides better inter-cluster distance than other techniques except wine dataset. For wine dataset, IHSC gives well-separated clusters.

Table 7. Mean and standard deviation of intra and inter-cluster distance Measures for Wine Dataset

	K-Means	GAC	HSC	IHSC	MHSC
Inter-cluster distance	265.78702 (8.14389)	333.30569 (18.23820)	332.81698 (7.69306)	<b>333.12396</b> <b>(7.49648)</b>	332.76229 (9.57703)
Intra-cluster distance	131.81582 (2.82912)	101.64894 (4.51370)	99.42088 (3.34328)	100.13992 (2.59244)	<b>98.72892</b> <b>(2.76578)</b>

Table 8. Mean and standard deviation of intra and inter-cluster distance Measures for Haberman Dataset

	K-Means	GAC	HSC	IHSC	MHSC
Inter-cluster distance	3.11350 (0.19132)	8.89654 (0.14653)	9.06112 (0.29913)	8.94582 (0.16251)	<b>9.0589</b> <b>(0.16805)</b>
Intra-cluster distance	11.13102 (0.03320)	8.59276 (0.08134)	8.47006 (0.12799)	8.57177 (0.09103)	<b>8.48354</b> <b>(0.06714)</b>

Table 9 Mean and standard deviation of intra and inter-cluster distance Measures for Bupa Dataset

	K-Means	GAC	HSC	IHSC	MHSC
Inter-cluster distance	35.13750 (0.00000)	50.22612 (9.99791)	49.48996 (4.64680)	51.83883 (6.09530)	<b>53.46400</b> <b>(2.41748)</b>
Intra-cluster distance	42.95460 (0.00000)	41.26448 (2.14356)	<b>40.8574</b> <b>(1.22492)</b>	41.42587 (1.61761)	<b>41.10058</b> <b>(1.19719)</b>

Table 10. Mean and standard deviation of intra and inter-cluster distance Measures for CMC Dataset

	K-Means	GAC	HSC	IHSC	MHSC
Inter-cluster distance	1.69890 (0.32438)	8.61587 (0.72308)	8.85905 (2.01719)	8.79613 (0.25072)	<b>8.85460</b> <b>(0.19133)</b>
Intra-cluster distance	7.50016 (0.13887)	3.8874 (0.06174)	3.97840 (0.12658)	3.85730 (0.01722)	<b>3.84178</b> <b>(0.00797)</b>

Table 11. Mean and standard deviation of intra and inter-cluster distance Measures for Breast Cancer Dataset

	K-Means	GAC	HSC	IHSC	MHSC
Inter-cluster distance	6.68060 (0.00000)	<b>7.00333</b> <b>(0.05034)</b>	6.95620 (0.03055)	6.94716 (0.03221)	<b>6.97342</b> <b>(0.03220)</b>
Intra-cluster distance	5.31780 (0.00000)	5.39902 (0.13399)	5.28189 (0.02979)	5.27572 (0.02787)	<b>5.26862</b> <b>(0.03116)</b>

## 6. Conclusion

In this paper, a modification has been proposed in the improved harmony search algorithm, which is based on fine-tuning of two important parameters HMCR and PAR. The impacts of these constant parameters have been evaluated. The proposed algorithm has been implemented and tested on five real life datasets. On comparing the results of proposed technique with the others, it has been found that MHSC resulted in better cluster quality metrics than HSC, IHSC, GAC techniques and K-Means. Experimental results demonstrate that the MHSC yields better intra-cluster distance than other techniques for all the datasets except the wine dataset. It has been experimentally validated that MHSC provides highly cohesive clusters than the other techniques.

## References

- Yang, X-S., 2010. Nature-inspired metaheuristic algorithm. 2nd ed. Luniver Press.
- Osman, I.H., Laporte, G., 1996. Metaheuristics: A bibliography, *Annals Operations Research* 63, p. 513–623.
- Blum, C., Andrea, R., 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Computing Surveys* 35(3), p. 268–308.
- Yagiura, M., Ibaraki, T., 2001. On metaheuristic algorithms for combinatorial optimization problems, *Systems and Computers in Japan* 32, p. 33–55.
- Omran, M.G.H., Mahdavi, M., 2008. Global-best harmony search, *Journal of Applied Mathematics and Computation* 198, p. 643–656.
- Geem, Z.W., Kim, J., Loganathan, G.V., 2001. A new heuristic optimization algorithm: harmony search, *Journal of Simulation* 76(2), p. 60–68.
- Mahdavi, M., Fesanghary, M., Damangir, E., 2007. An improved harmony search algorithm for solving optimization problems. *Journal of Applied Mathematics and Computation* 188(2), p. 1567–1579.
- Malaki, M., Pourbagheri, A., Abolhassani, H., 2008. "A combinatory approach to fuzzy clustering with harmony search and its applications to space shuttle data," In *Proc. Soft Comput. Intell. Sys.* 2008; 1154–1159.
- Jain, A.K., Murthy, M.N., Flynn P.J., 1999. Data clustering: a review. *ACM Computing Surveys* 31, p. 264–323.
- Xu, R., Wunsch, D.C., 2009. *Clustering*. John Wiley and Sons, Inc, NJ, Hoboken.
- Amiri B, Hossain L, Mosavi SE. Application of harmony search algorithm on clustering. In *Proc. World Congress Engg. Comp. Sci.* 2010; 460–465.
- Kowalski, G. *Information retrieval systems: theory and implementation*. Kluwer Academic Publishers; 1997.
- Buckland, M., Gay, F., 1994. The relationship between recall and precision. *Journal of American Society for Information Science* 45(1), p. 12–19.
- Worasuchep, C., 2011. A harmony search with adaptive pitch adjustment for continuous optimization. *Journal Hybrid Infomation Technology* 4(4), p. 13–23.
- Blake, C. L., Merz, C. J., 1998. UCI Repository of Machine Learning Databases  
<http://www.ics.uci.edu/~mllearn/databases/>.