IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 13, 2017, accepted March 31, 2017, date of publication April 5, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2691412

A New Heuristic for the Data Clustering Problem

UMAIR F. SIDDIQI[1], (Member, IEEE), and SADIQ M. SAIT[2], (Senior Member, IEEE)
[1]Center of Communications and IT Reserach, Research Institute, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[2]Department of Computer Engineering and the Center of Communications and IT Research, Research Institute, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Sadiq M. Sait (sadiq@kfupm.edu.sa)

ABSTRACT This paper presents a new heuristic for the data clustering problem. It comprises two parts. The first part is a greedy algorithm, which selects the data points that can act as the centroids of well-separated clusters. The second part is a single-solution-based heuristic, which performs clustering with the objective of optimizing a cluster validity index. Single-solution-based heuristics are memory efficient as compared with population-based heuristics. The proposed heuristic is inspired from evolutionary algorithms (EAs) and consists of five main components: 1) genes; 2) fitness of genes; 3) selection; 4) mutation operation; and 5) diversification. The attributes of the centroids of clusters are considered as genes. The fitness of a gene is a function of two factors: 1) difference between its value and the same attribute of the mean of the data points assigned to its cluster and 2) the frequency with which it has been mutated in previous iterations. The genes that have low fitness values should be updated through the mutation operation. The mutation operation performs small change (positive or negative) in the value of the gene. The mutants are accepted if they are better (with respect to objective function) than their parents. However, diversification in the search process is maintained by allowing, with a small probability, the mutants to replace their parents even they are not better than them. The objective functions used in the proposed heuristic are Calinski Harabasz index and Dunn index. The proposed algorithm has been experimented using real-life numeric data sets of UCI repository. The number of data points and number of attributes in the datasets lie between 150–11 000 and 4–60, respectively. The results indicate that the proposed algorithm performs better than two standard EAs: 1) simulated annealing algorithm and 2) differential evolution algorithm and a genetic algorithm-based clustering method.

INDEX TERMS Data clustering, simulated evolution, applications of evolutionary algorithms, genetic algorithm.

I. INTRODUCTION

Clustering refers to the partitioning of a set of data-points into groups in such a way that each data-point is maximally similar to the data-points within its cluster [1], [2]. Clustering is an important problem in data-mining and machine learning. Some popular applications of clustering are as follow: (i) It is used to summarize data in many data-mining problems such as outlier analysis and classification; (ii) It is used to group like-minded users and similar customers in collaborative filtering and customer segmentation; (iii) It is used to create compact data representation; (iv) It is used to detect key trends and events in the streaming data of social networking applications; and (v) It is used to group similar genes in gene-expression data analysis [2], [3]. The clustering problem is NP hard when the number of clusters is more than three [4].

Clustering algorithms are usually classified into two types: (a) Partitional clustering, and (b) Hierarchical clustering. Partitional clustering algorithms iteratively splits data into clusters. A data-item can belong to only one partition. The total numbers of clusters (K) should be known in advance, unless, additional methods are employed to determine the number of clusters. In hierarchical clustering, a dendrogram (or clustering tree) is generated. The first step is to build a similarity matrix between all data-points and selects a pair of data-items that are maximally similar to each other. In the second step, the similarity matrix is updated and the data-items that were selected in the previous step are replaced by a single entry for the pair. The remaining steps repeat the same procedure to complete tree construction [5]. Hierarchical clustering automatically determine the number of clusters.

The quality of clustering is measured in terms of its compactness and separation. A cluster is said to be compact when its data-points are similar to each other. A cluster has good separation when its data-points are maximally dissimilar with the data-points of the other clusters. The similarity between two data-items can be determined in terms of several

measures such as as: Minkowski Distance, Cosine distance, Correlation coefficients (e.g. Pearson, Spearman). Minkowski Distance is the most popular method and as a parameter p. When $p = 1$, it yields Manhattan distance, and when $p = 2$, it returns Euclidean distance. The choice of similarity measure usually depends on the application area where clustering is applied. Euclidean distance is most commonly used similarity measure and produces good results in majority of applications [6]. The quality of a clustering solution is determined using a validity index. The validity indices compute both compactness and separation between clusters. Some popular quality measures are as follows: (a) Davies Bouldin Index (DBI) [7]; (ii) Calinski Harabasz Index (CHI) [8]; (iii) Dunn Index (DI) [9], [10]; (iv) Silhouette Index (SI) [11]; and (v) SD Validity Index (SDI) [12]. In this work we used CHI and DI.

In optimization perspective, clustering problem is considered as an NP-hard grouping problem [13], [14]. Heuristics such as Evolutionary algorithms (EAs) are popular in solving NP-hard problems [15], [16]. Recently, several evolutionary algorithms (EAs) have been proposed to perform clustering. The EAs can perform clustering using either a fixed or variable K value and find clustering that is optimal w.r.t. to a validity index. The EAs with a fixed K value are useful in the following two cases: (i) Some information about the classes in data is known, or (ii) The value of K can be obtained using other methods such as the method proposed by Sugar and James [17]. The EAs are compared with each other in terms of two criterion: (i) their best objective function value; and (ii) the number of evaluations of the objective function they need to converge to their best result (known as evaluation count or number of evaluations). The objective function is usually computationally intensive and the EAs that have a large evaluation count are considered to be slower than the EAs that have a smaller evaluation count [18]–[21]. The EAs can either use a population of solutions or use only one solution. The single-solution-based EAs have smaller evaluation count but their solution quality is usually not as good as population-based EAs.

This article proposes a new heuristic for the clustering of numeric data and the objective is to maximize CHI or DI. The proposed heuristic consists of two parts. The first part is a greedy algorithm which selects the data-points that can act as centroids of clusters and the criterion is to maximize the separation between clusters. The second part is a single-solution based heuristic whose components are functions from Genetic Algorithm (GA) and Simulated Evolution (SimE) algorithm [15]. The attributes of the centroids of the clusters are considered as genes. The heuristic finds optimal clusters by determining optimal values of all genes w.r.t. a cluster validity index. In each iteration, the fitness of all genes is determined and the genes of lesser fitness values go through the mutation operation. The selection of genes for mutation resembles the creation of selection set for the allocation operation in the SimE algorithm. The mutants that improve the objective function value of the solution always replace their parents, whereas, the remaining mutants only replace their parent with small but variable probability. The iterations continue until the stopping criterion (maximum runtime or maximum iterations) is reached. Experiments have been conducted to compare the proposed heuristic with two standard EAs: (i) Simulation Annealing (Gen-SA) [22]; and (ii) Differential Evolution (DE) [23] and a Genetic Algorithm (GA) for the clustering problem [24]. The real-life datasets of the UCI repository [25], [26] have been used in the experiments. The analysis of the experimental results show that the proposed heuristic is better than the other heuristics in terms of its solution quality and number of evaluations to reach optimal value.

This paper is organized as follows. The second section briefly describes some existing EAs for the clustering problem. Third section describes the clustering problem. Fourth section describes proposed heuristic. The experimental results are presented in the fifth section. The last section contains the conclusion.

II. PREVIOUS WORK

This section describes some EAs for the clustering problem. Selim and Alsultan proposed an application of Simulated Annealing (SA) algorithm to the clustering problem [27]. The solution is represented in terms of an assignment vector of length equal to the number of data-points. For each data-point, the vector holds the index of the cluster to which it is currently assigned. The perturb operation consists of changing the assignment of a randomly selected data-point. The solution obtained from the perturb operation is always accepted if it is better than the existing one, otherwise, it is accepted with a very small probability.

Maulik and Bandyopadhyay proposed a Genetic Algorithm (GA) for the clustering problem [24]. The chromosome is represented by a vector that contains centroids of all clusters. The objective function is equal to the sum of the Euclidean distances of the data-points from the centroids of their clusters. The fitness of a centroid (or cluster) is computed in two steps. In the first step, the centroid is updated to the current mean of the data-points that are assigned to it. The second step is to compute the mean of the Euclidean distances of all data-points from the centroids of their clusters. The selection function uses fitness values to select the best chromosomes from the population. It uses one-point crossover and mutation operations and fixed cross-over and mutation probabilities. In the mutation operation, an attribute is randomly selected and a random number between 0–1 is added or subtracted to it. The experimental results showed that the GA-based clustering method has produced much better results as compared to the K-means method.

Das et al. have proposed a Differential Evolution (DE) algorithm for the clustering problem that also automatically determines the number of clusters [18]. The chromosome consists of two portions. The first portion stores the activation thresholds of clusters and the second portion stores the centroids of clusters. A cluster is considered active if

its activation threshold is greater than a pre-defined value (e.g. 0.5). The fitness of a chromosome is equal to the reciprocal of a cluster validity metric such as Davies Bouldin index (DBI). In each iteration, the data-points are assigned to their nearest active clusters. The DE algorithm creates a new generation of chromosomes by updating the centroids or activation thresholds of the clusters. Changes in the centroids and/or active thresholds values of a chromosome could lead to a new clustering solution. The algorithm ensures that in any chromosome, at-least two clusters should remain active. The experimental results showed that it can perform better than some existing algorithms such as GA-based clustering [24] and standard DE algorithm.

Kang et al. have proposed a clustering algorithm based on K-means and Mussels wandering optimization (MWO) [28]. The MWO basically overcomes the shortcomings of the K-means method. In MWO a solution is called a mussel and contains the centroids of all clusters. The sum of squared errors (SSE) metric is used as the fitness function of a mussels. Each iteration of the MWO algorithm consists of the following three steps: (i) A small pre-defined number of mussels which have best fitness values are determined and their center is calculated; (ii) The position of the mussels are updated following the procedure used in the MWO and with the help of the center calculated in the previous step; and (iii) At the end of each iteration, the top mussels are redetermined and a new center is calculated for the next iteration. The experiments indicate that the algorithm performed better than K-means and a hybrid of K-means with particle swarm optimization (PSO) algorithm.

III. PRELIMINARY CONCEPTS

This section presents some relevant preliminary concepts and definitions. Consider a data set D that contains N data points and is represented by $D = \{d_0, d_1, \ldots, d_{N-1}\}$. Each data-point $d_i \in D$ consists of m attributes and represented by $d_i = \{x_0, x_1, \ldots, x_{m-1}\}$, where $x_i \in R$. A partitional clustering algorithm tends to find a set of K clusters represented by $\{C_0, C_1, \ldots, C_{K-1}\}$. A cluster C_j is represented by two terms (i) its centroid ($C_j^c = \{c_0, c_1 \ldots, c_{m-1}\}$), and (ii) the data-points which are assigned to it ($C_j^p = \{p_0, \ldots, p_{n_j-1}\}$), where n_j represents the number of data-points that are assigned to C_j. Any attribute of p_i is represented by $p_i[x_j]$, where $j \in \{0, 1, \ldots, m-1\}$ and indicates the index of the attribute. Any two clusters cannot have a same centroid (i.e., $C_j^c \neq C_k^c$, for $j \neq k$). The assignment of data-points to any cluster C_j should meet the following condition: $C_0^p \cup C_1^p \ldots \cup C_{K-1}^p = D$. The center of all data-points in D is represented as C. The centroid of a cluster is equal to the means of all data-points that are assigned to it (assuming that the similarity measure is Euclidean distance). Many clustering algorithms including this work try to find optimal centroids of the clusters rather than finding optimal assignment of data-points. Given a set of centroids, the data-points are assigned to the cluster whose centroid is nearest to it or maximally similar to it using a similarity measure. Euclidean distance is the most commonly used similarity measure and is used in this work. The Euclidean distance between two data-points d_i and d_j is represented by $||d_i - d_j||$.

Many cluster validity indices have been developed to measure the quality of clustering. This work uses two well-established validity indices which are as follows: (a) Calinski Harabasz index (CHI) [8], and (b) Dunn index (DI) [10]. Both indices compute the ratio of the separation of clusters to their compactness. CHI is defined in (1). The term in numerator computes the average of the squared distance between the centroids of different clusters (C_k^c) and the global center of the data-points (C). The term in denominator computes the averaged squared distance of the data-points from the centroids of their clusters. The maximum value is desirable and refers to well-separated and compact clustering.

$$\text{CHI} = \frac{\frac{\sum_{k=0}^{K-1} n_k ||C_k^c - C||^2}{K-1}}{\frac{\sum_{k=0}^{K-1} \sum_{d_j \in C_k} ||d_j - C_k^c||^2}{N-K}} \tag{1}$$

The DI is the ratio of the minimum distance between any two data-points that belong to different clusters to the maximum distance between any two-points that lie in a same cluster. The DI is defined in (2), (3), and (4). The function '$\delta(u, v)$' is the smallest distance (or Euclidean distance) between any two data-points that belongs to two different clusters u and v. The function '$\Delta(w)$' is the largest distance between any two data-points that belongs to a same cluster i.e., C_w (where C_w^p is the set of all data-points which are assigned to C_w). DI is determined as the ratio of the smallest value of $\delta(u, v)$ over all possible values of u and v (provided $u \neq v$) to the largest value of $\Delta(w)$. A bigger value of DI means better clustering.

$$\delta(u, v) = \min_{x \in C_u^p, y \in C_v^p} (||x - y||) \tag{2}$$

$$\Delta(w) = \max_{\{x,y\} \in C_w^p} (||x - y||) \tag{3}$$

$$\text{DI} = \frac{\min_{\{u,v\} \in \{0..K-1\}, u \neq v} (\delta(u, v))}{\max_{w \in \{0..K-1\}} (\Delta(w))} \tag{4}$$

IV. PROPOSED HEURISTIC

This section describes the proposed heuristic in detail. Fig. 1 shows the main components of the proposed clustering heuristic. It consists of two parts. The first part is a greedy algorithm whose aim is to find points from the data-set that can act as centroids of clusters. The criteria for the selection of centroids is to maximize the inter-cluster separation. The second part is a heuristic that contains some features of GA and SimE algorithms and performs clustering by optimization. The objective function of optimization is a validity index that considers both separation as well as compactness of clusters. The input of the proposed heuristic consists of the following items: (i) Set of data-points (D); (ii) Number of clusters (K), (iii) Five parameters ($\alpha, \beta, \delta, p_m, B$). The first two parameters (α and β) belong to the first part and the

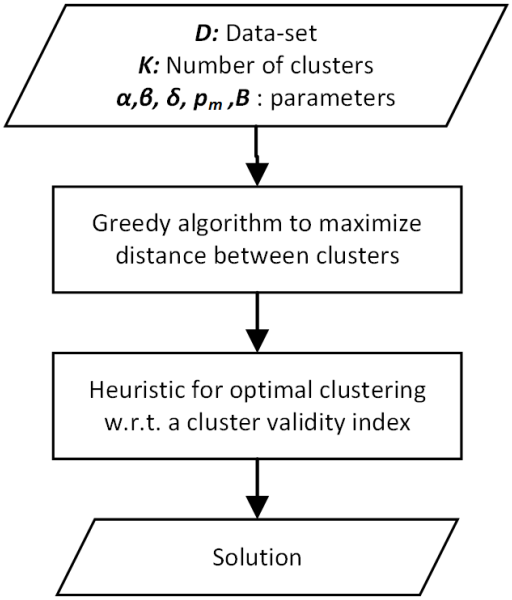


FIGURE 1. Main components in the proposed heuristic for clustering.

remaining three parameters belong to the second part of the heuristic.

A. OBJECTIVE FUNCTION

The objective function used in both algorithms is a validity index that considers both separation and compactness of clusters. The objective function is represented by f_n and its possible values are $f_n \in \{\text{CHI, DI}\}$. The validity indices CHI and DI are already described in (1) and (4). The values of both indices should be maximized in the optimization.

B. ALGORITHM FOR FINDING OPTIMAL CENTROIDS AS DATA-POINTS

Fig. 2 shows the first part of the algorithm. All the inputs of the algorithm are already described at the start of Section IV. The parameters α and β are related to the stopping criterion of the algorithm. In line 1 of Fig. 2, up-to K data-points have been selected as centroids. In line 4, The set $\overline{D}$ holds the data-points which are not currently acting as centroids. In line 6, the set D_z holds the centroids of all clusters except the $z+1^{th}$ cluster (the cluster C_z is the $z+1^{th}$ cluster because the indices of clusters starts from zero.) The set P_z stores a copy of the centroid of the $z+1^{th}$ cluster. In line 7, f_0 is the value of the objective function before any change has taken place in the current iteration. In line 8, a data-point is chosen as the new centroid of the $z+1^{th}$ cluster. As the equation shows that the new data-point should be the one which has maximum distance from the centroids of the remaining clusters. In line 9, the values of the objective function before and after the change are compared and the new centroid will be discarded if it worsens the value of the objective function. Line 13 contains a condition to terminate the loop if the last β iterations are unable to produce any change in the centroids. The algorithm can execute for up-to α number of iterations.

C. PROPOSED HEURISTIC FOR THE CLUSTERING PROBLEM

Fig. 3 shows the proposed heuristic which is used in the second part of the data-clustering method. The initial solution comprises of the centroids determined by the greedy algorithm. Each attribute of the centroid is considered as a gene. In step 2, the fitness of all genes is computed. In step 3, a selection set is prepared that contains the genes that have low fitness values, however, genes of high fitness values could also be selected with a small probability. In step 4, the mutation operation is applied to the selected genes. In step 5, the mutants refer to the new values of the genes obtained by applying the mutation operation. The mutants replace their parents (i.e., the existing values of genes) if they do not worsen the objective function value. However, the mutants that can worsen the objective function value can also accepted with a very small probability. The iterations proceeds until the stopping criterion is reached. The different steps are described below in detail.

1) STEP 1: INITIALIZATION

In this step, the centroids determined by the greedy algorithm are set as the initial solution. The centroids are represented as $\{C_0^c, C_1^1, \ldots, C_{K-1}^c\}$ and the attribute of a centroid C_j^c are represented by $\{c_0, c_1, \ldots, c_{m-1}\}$.

2) STEP 2: FITNESS COMPUTATION

In this step, the fitness of the attributes of all centroids is determined. The fitness computation is based on the principle of K-means method, i.e., in each iteration, the centroids of clusters are assigned equal to the mean of the data-points that are assigned to them. In the proposed heuristic, the fitness of an attribute is inversely proportional to two quantities: (a) the difference of that attribute from the same attribute of the mean of the data-points, and (ii) the number of times that attribute has been mutated in previous iterations. Equations (5), (6), (7) and (8) show the computation of fitness values of all attributes (i.e., m attributes) of centroid C_j^c (which is the centroid of the j^{th} cluster). In (5), the mean of the data-points that are assigned to the j^{th} cluster is computed and represented by C_j^m (C_j^m has m attributes). The term $\sum_{i=0}^{n_j-1} p_i[x_0]$ refers to the sum of the first attribute to all data points that are assigned to the j^{th} cluster. The total number of data-points assigned to the j^{th} cluster is equal to n_j. In (6), a difference is computed between the current centroid value of the j^{th} cluster (C_j^c) and the mean value from (5). In (6), point-wise differences are computed between the same attributes. In (7), the difference values are divided by the history of the attributes. The history of an attribute is the number of times it has been mutated in previous iterations. The calculation are again point-wise and the difference of the k^{th} attribute is divided by the history of the k_{th} of centroid C_j^c. In (8), the

Input: D, K, α, β, f_n
Output: $\{C_0^c, ...C_{K-1}^c\}$

Initialize the centroids of K clusters (i.e., $\{C_0^c, C_1^c, ..., C_{K-1}^c\}$) to randomly selected unique points from D.
i = 0
while $i < \alpha$ **do**
  $\overline{D} = D - \{C_0^c, C_1^c, ..., C_{K-1}^c\}$
  z= a random number between 0 and K-1
  D_z= $\{C_0^c, C_1^c, ..., C_{K-1}^c\} - C_z^c$, P_z= C_z^c
  $f_0 = f_n(\{C_0^c, ..., C_{K-1}^c\})$
  $C_z^c = \underset{y \in \overline{D}}{argmax}(\frac{\sum_{x \in D_z}(||x-y||)}{K-1})$
  if $fn(\{C_0^c, ..., C_{K-1}^c\}) < f_0$ **then**
    $C_z^c = P_z$
  endif
  i++
  if no change in $\{C_0^c, ...C_{K-1}^c\}$ occurs in the last β iterations **then**
    Break
  endif
endwhile
return $(\{C_0^c, ...C_{K-1}^c\})$

FIGURE 2. Proposed algorithm for finding centroids as data-points.

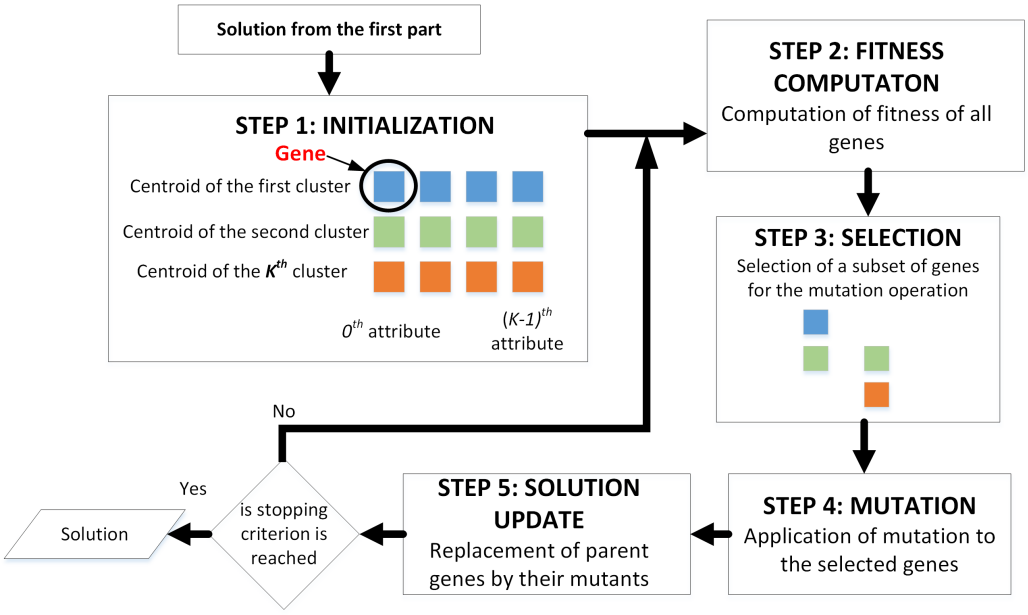


FIGURE 3. Proposed heuristic for data clustering.

values are normalized which are termed as fitness values. The vector f_j consists of m attributes and any k^{th} attribute of f_j holds the fitness value of the k^{th} attribute of C_j^m.

$$C_j^m = \frac{1}{n_j}\{\sum_{i=0}^{n_i-1} p_i[x_0], \ldots, \sum_{i=0}^{n_i-1} p_j[x_{m-1}]\} \quad (5)$$

$$\Delta_j = C_j^c - C_j^m \quad (6)$$

$$V_j = \frac{\Delta_j}{H[j]} \quad (7)$$

$$f_j = \frac{V_j}{\max(V_j)} \quad (8)$$

3) STEP 3: SELECTION

The proposed heuristic uses the selection function of the SimE algorithm [15] and fitness value is used in place of the goodness value. The selection function uses a

parameter B which is the Bias factor and its value could lie between $[-0.2, +0.2]$. The selection function is described in (9). The function applies the selection function on the j^{th} attribute of centroid C_i^c and the result could be 1 or 0. The term 'Random' indicates a random number between [0,1]. The attributes whose result from the selection function is one should go through the mutation operation.

$$s_{ij} = \begin{cases} = 1 & \text{if Random } < 1 - f_i[j] + B \\ = 0 & \text{otherwise} \end{cases} \tag{9}$$

4) STEP 4: MUTATION

The mutation operation is applied to an attribute at a time and it can make a small change in its value. The steps in the mutation operation for the j^{th} attribute of C_i^c are as mentioned below. The existing value of the j^{th} attribute is represented by c_j and the value after the mutation operation is represented by c'_j.

1) Determine the lower (l_j) and upper (u_j) bounds for the j^{th} attribute as mentioned in (10) and (11). The lower and upper bounds are equal to the minimum and maximum values of the j^{th} attributes of all points in the data-set D.
2) Compute two intermediate terms: t_l and t_u, where $t_l = \frac{c_j - l_j}{2}$ and $t_u = \frac{u_j - c_j}{2}$.
3) The new value of the j^{th} attribute (i.e., c'_j) is equal to a randomly selected value from a uniform distribution between $c_j - t_l$ and $c_j + t_u$.

$$l_j = \min_{d_i \in D}(d_i[j]) \tag{10}$$

$$u_j = \max_{d_i \in D}(d_i[j]) \tag{11}$$

5) STEP 5: SOLUTION UPDATE

The value of an attribute obtained from the mutation operation always replace the existing value of that attribute if it does not worsens the objective function value. Otherwise, it is accepted only with a very small probability. The procedure to accept a mutant is mentioned below. The existing value of the attribute is represented by c_j and the mutant value is represented by c'_j.

1) Compute f_0 as equal to the objective function value when the j^{th} attribute has value equal to c_j.
2) Compute f_1 as equal to the objective function value when the j^{th} attribute has value equal to c'_j.
3) Compute $\Delta = \frac{||f_0 - f_1||}{f_0}$
4) If f_1 is better than or equal to f_0 then accept the mutant, i.e., $c_j = c'_j$
5) Otherwise, accept the mutant under the following two conditions: (a) $\Delta \leq \delta$, and (ii) with acceptance probability equal to p_m.

Both parameters δ and p_m are real numbers between [0, 1]. The acceptance of worse solutions increases the diversity in the search process, however, the values of δ and p_m should be kept very small in-order to avoid random walk like behavior. The trapping of search into local optima can also be avoided with the help of acceptance of bad moves.

V. EXPERIMENTAL RESULTS

The proposed heuristic has been implemented and executed using R version 3.3.2 on a Linux-based system. The parameter values used are as follows: $\alpha = 300$, $\beta = 10$, $\delta = 0.01$, $p_m = 0.01$, and $B = -0.2$. The parameter values have been determined on the simplest problem 'iris' through trial and error using some possible values. The dataset of real-life problems from the UC Irvine machine learning repository [25], [26] have been used in the experiments. The benchmarks have only numeric attributes and have been previously used in the evaluation of the clustering algorithms such as Swarm intelligence and Differential evolution based clustering methods [18], [28]. Table 1 shows the important characteristics of the benchmarks. The number of data-points lie from 150–10992, number of attributes lie between 4–60 and number of classes in the data lie between 2–10. The experiments consists of two parts. The first part considers the CHI validity index as the objective function and the second part uses DI validity index as the objective index.

TABLE 1. Characteristics of real-life data-sets.

Problem	# of data-points	# of clusters (K)	# of attributes
Iris	150	3	4
Glass	214	6	9
Ecoli	336	8	7
Banknote authentication	1372	2	4
Image segment	2310	7	19
Cardiotocography	2126	10	21
Student evaluation	5820	3	32
Landsat satellite	6435	6	36
Pen-based digits	10992	10	16
Balance scale	627	4	4
Diabetes	769	3	8
Heart-statlog	271	3	13
Ionosphere	352	3	34
Sonar	209	3	60
Vehicle	847	5	18
Waveform-500	5001	4	40

The performance of the proposed heuristic has been compared with three existing algorithms which are as follows: (a) standard Simulated Annealing (Gen-SA) [22]; (b) standard Differential Evolution (DE) [23]; and (c) Genetic Algorithm for clustering (GA) [24]. The Gen-SA and DE algorithms are available as packages in R. The GA algorithm has been implemented in R according to its description [24]. The Gen-SA and DE algorithms have been executed with standard parameter values. The GA algorithm has been executed with the same parameter values as used by its authors [24], i.e., mutation probability= 0.001, cross-over probability= 0.80, and population-size= 100.

The non-deterministic nature of the algorithms has been considered by conducting up-to 50 trials on each problem. The termination condition of the Gen-SA, DE and GA was set as equal to twice of the maximum number of evaluations of the proposed heuristic in any trial to solve the same problem. For example, if the maximum number of evaluations of the

TABLE 2. Solution quality results when objective function is to maximize CHI [8].

Problem	Proposed		Gen-SA		DE		GA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Iris	**561.6278**	0	561.5296	0.4915	**561.6278**	0	**561.6278**	0
Glass	**124.0103**	2.2869	123.4779	1.7761	112.1484	3.509	117.1415	2.6268
Ecoli	**145.6411**	3.1929	143.2815	0	108.4265	4.9368	138.8787	2.9514
Banknote authentication	1423.4569	0.3172	1423.4145	0.5004	**1423.4756**	0.1685	1409.4623	9.7308
Image segment	**1071.86**	33.5375	974.6373	21.7361	641.4499	60.3651	1001.1856	17.7872
Cardiotocography	**722.9805**	12.5134	682.0008	1.8994	466.5368	20.7833	683.592	7.2119
Student evaluation	3204.1164	82.9119	**3211.0215**	34.0916	2955.5275	38.4589	3198.1964	13.901
Landsat satellite	4646.7676	211.9314	4711.088	178.8783	3537.1646	81.9229	**4719.3496**	26.0458
Pen-based digits	2734.8316	67.1278	**2757.1396**	56.1514	1902.3729	73.1081	2590.8703	61.5258
Balance scale	135.6768	0.5193	**135.9312**	0.5975	133.3655	0.6095	129.9785	2.1337
Diabetes	589.9591	0.8942	**589.9601**	0.9465	587.6764	1.1668	585.2821	0.7508
Heart-statlog	181.0285	2.5323	**181.5804**	0.3113	180.4245	0.5803	179.3031	0.3113
Ionosphere	**93.5787**	2.4035	93.5308	2.4206	86.3052	1.2365	88.9108	1.3794
Sonar	**34.6654**	0	34.6564	0.0254	29.447	0.7991	25.6406	0.7364
Vehicle	1538.6454	4.1843	**1534.7837**	8.3351	1488.0946	12.4833	1527.8957	3.3607
Waveform-5000	635.2917	20.8169	**645.5267**	8.2394	513.7847	11.6154	570.2637	9.661

TABLE 3. Number of evaluations when objective function is to maximize CHI [8].

Problem	Proposed		Gen-SA		DE		GA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Iris	**353**	519	975	879	1749	1102	655	1359
Glass	**4889**	2999	13761	2427	22480	3159	31884	20123
Ecoli	**7535**	3797	25763	0	23430	2781	27962	25973
Banknote authentication	**734**	462	1511	860	2295	646	2801	2974
Image segment	**21140**	9586	61405	3964	56495	7672	28068	37426
Cardiotocography	**43957**	7887	100415	301	94763	6969	139412	73565
Student evaluation	**16226**	4817	39176	4512	41929	2719	71098	31612
Landsat satellite	**20574**	4964	47522	1519	44864	5325	38417	40614
Pen-based digits	**28497**	6112	64319	1789	63498	4098	99891	47440
Balance scale	**1765**	876	5043	1102	5720	991	7968	6045
Diabetes	**3427**	1533	7917	2474	10204	1263	18696	9442
Heart-statlog	**3472**	3024	7191	4818	14502	3257	23931	15531
Ionosphere	**10243**	7341	27275	9204	44339	4472	10504	25780
Sonar	6790	4970	37461	5719	44079	4346	**3040**	7955
Vehicle	**13783**	6082	31861	8704	39150	3245	55964	30373
Waveform-5000	17636	2471	41410	1670	38967	5006	**785**	642

proposed heuristic in the fifty trials of 'iris' is 100, then the other algorithms have been executed for up-to 200 number of evaluations on the 'iris' problem. The results of different algorithms are compared using the average value of their trials and with the help of t-tests [29]. T-tests are commonly used to compare two or more EAs [28], [30].

A. WHEN THE OBJECTIVE FUNCTION IS TO MAXIMIZE CHI

Table 2 shows the CHI values of the proposed and other algorithms. The results of each algorithm are presented under its label and consists of two columns. The first column ('Mean') contains the mean value of the fifty trials and the second column ('SD') contains the standard deviation of the fifty trials. The results indicate that the mean CHI values of the proposed heuristic is better than the other algorithms in most of the problems. Table 3 shows the number of evaluations consumed by the algorithms in obtaining their best results. The results of each algorithm consists of two columns. The first column contains the mean and the second column contains the standard deviation. The results show that the proposed heuristic requires very small number of evaluations to reach its best results as compared to the other algorithms.

Tables 4 and 5 show the results of the two-sided t-tests [29] to determine if the solution quality (CHI) and number of evaluations of the proposed heuristic are better than the other algorithms. The t-tests have been performed with significance level equal to 0.05. A t-test compares results of two algorithms at a time and returns a p-value. When the p-value is equal to or greater than the significance level (0.05) then the results of both algorithms are considered equal to each other. However, when the p-value is smaller than the significance level then the results of the two algorithms are not equal and the algorithm that has a better mean is considered better. Tables 4 and 5 also contains a column 'remarks', that indicates if the result of the proposed heuristic is equal, better or worse than the other algorithm.

A comparison of the proposed heuristic with Gen-SA using the results in Table 4 suggests the following: (i) the proposed heuristic produced better results in five problems; (ii) the results are equal in eight problems; and (iii) the results of

TABLE 4. Comparison the CHI [8] results of the proposed heuristic with others using t-tests.

Problem	vs. Gen-SA		vs. DE		vs. GA	
	p-value	remarks	p-value	remarks	p-value	remarks
Glass	0.1968	Equal	8.763e-34	Better	7.885e-25	Better
Ecoli	3.553e-06	Better	2.512e-60	Better	9.076e-19	Better
Banknote authentication	0.6143	Equal	0.713	Equal	1.142e-13	Better
Image segment	2.928e-29	Better	3.443e-56	Better	3.913e-21	Better
Cardiotocography	1.408e-28	Better	4.533e-76	Better	1.783e-31	Better
Student evaluation	0.5879	Equal	1.83e-29	Better	0.6206	Equal
Landsat satellite	0.1368	Equal	4.327e-36	Better	0.03311	Worse
Pen-based digits	0.07465	Equal	3.445e-78	Better	3.766e-19	Better
Balance scale	0.0253	Worse	1.296e-36	Better	4.657e-25	Better
Diabetes	0.9957	Equal	2.092e-18	Better	3.695e-48	Better
Heart-statlog	0.1324	Equal	0.106	Equal	1.545e-05	Better
Ionosphere	0.9211	Equal	4.582e-30	Better	3.048e-19	Better
Sonar	0.01503	Better	4.698e-42	Better	2.776e-55	Better
Vehicle	0.004563	Better	2.359e-35	Better	5.194e-25	Better
Waveform-5000	0.0004974	Worse	1.271e-62	Better	3.886e-38	Better
Note: significance level = 0.05						

TABLE 5. Comparison of the number of evaluations of the proposed heuristic with others using t-tests when objective function is CHI [8].

Problem	vs. Gen-SA		vs. DE		vs. GA	
	p-value	remarks	p-value	remarks	p-value	remarks
Iris	4.701e-05	Better	1.253e-11	Better	0.148	Equal
Glass	4.671e-29	Better	3.154e-49	Better	1.046e-12	Better
Ecoli	1.052e-35	Better	1.127e-40	Better	1.213e-06	Better
Banknote authentication	2.991e-07	Better	5.494e-24	Better	1.168e-05	Better
Image segment	1.453e-37	Better	3.902e-36	Better	0.2101	Equal
Cardiotocography	4.775e-44	Better	1.116e-55	Better	3.141e-12	Better
Student evaluation	1.333e-43	Better	3.443e-47	Better	1.085e-16	Better
Landsat satllite	2.477e-35	Better	1.562e-35	Better	0.007168	Better
Pen-based digits	1.912e-43	Better	3.91e-51	Better	2.175e-14	Better
Balance scale	2.356e-29	Better	5.177e-38	Better	2.801e-09	Better
Diabetes	1.264e-17	Better	3.52e-42	Better	1.512e-15	Better
Heart-statlog	1.384e-05	Better	6.288e-32	Better	1.846e-12	Better
Ionosphere	6.42e-17	Better	1.861e-43	Better	0.9453	Equal
Sonar	7.7e-49	Better	1.006e-61	Better	0.005903	Worse
Vehicle	2.865e-20	Better	3.168e-39	Better	3.15e-13	Better
Waveform-5000	1.104e-87	Better	3.367e-49	Better	4.363e-58	Worse
Note: significance level = 0.05						

Gen-SA are better than that of the proposed heuristic in two problems. Table 4 also shows that the results of the proposed heuristic are better than that of DE in thirteen problems and equal to DE in two problems. The last two columns in Table 4 show that the results of the proposed heuristic are better than that of GA in thirteen problems, equal to GA in one problem and worse then GA in only one problem. Table 4 does not include the problem 'iris' because the results of iris are same in all trials (standard deviation is equal to zero for three algorithms) as shown in Table 2 and does not require further evaluation using t-tests. In 'iris' problem, all algorithms returned same results.

Table 5 shows the results of the t-tests that compare of the number of evaluations of the algorithms. The results convey the following information: (i) The number of evaluations of the proposed heuristic is better than that of Gen-SA and DE in all problems and better than that of GA in eleven problems.

Table 6 shows a summary of the results of t-tests to compare both solution quality (CHI) and number of evaluations (Eval. count). The results are expressed in terms of three symbols '$+, =, -$', which indicate that the proposed heuristic is better ($+$), equal ($=$) or worse ($-$) than the other algorithm. The results indicate that none of the other algorithms is better than the proposed heuristic in both solution quality and number of evaluations. When compared to Gen-SA, the proposed heuristic has same quality but better number of evaluations in majority of the problems. When compared to DE and GA, the proposed heuristic has better quality as well as number of evaluations in most of the problems.

B. WHEN THE OBJECTIVE FUNCTION IS TO MAXIMIZE DI

In the second part of experiments, the objective function is set to maximize cluster validity index DI. The results are presented in the same format as presented for CHI. Tables 7 and 8 present the solution quality (DI) and number of evaluations of the proposed and other algorithms. Tables 9 and 10 show the results of analysis using t-tests. The results in Table 9 convey the following information about the solution quality of the

TABLE 6. Summary of the comparisons using t-tests when objective function is CHI [8].

Problem	vs. Gen-SA		vs. DE		vs. GA	
	CHI	Eval. count	CHI	Eval. count	CHI	Eval. count
Iris	=	+	=	+	=	=
Glass	+	+	+	+	+	+
Ecoli	+	+	+	+	+	+
Banknote authentication	=	+	=	+	+	+
Image segment	+	+	+	+	+	=
Cardiotocography	+	+	+	+	+	+
Student evaluation	=	+	+	+	=	+
Landsat satellite	=	+	+	+	-	+
Pen-based digits	=	+	+	+	+	+
Balance scale	-	+	+	+	+	+
Diabetes	=	+	+	+	+	+
Heart-statlog	=	+	=	+	+	+
Ionosphere	=	+	+	+	+	=
Sonar	+	+	+	+	+	-
Vehicle	+	+	+	+	+	+
Waveform-5000	-	+	+	+	+	-

TABLE 7. Solution quality results when objective function is to maximize DI [10].

Problem	Proposed		Gen-SA		DE		GA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Iris	0.1665	0.006	**0.1691**	0	**0.1691**	0	0.1545	0.0072
Glass	**0.245**	0.0028	0.2447	0	0.2426	0.0058	0.1877	0.0122
Ecoli	**0.1494**	0.0151	0.135	0	0.1018	0.0101	0.1102	0.0137
Banknote authentication	0.0969	0.0132	0.1041	0.0017	**0.1043**	0.0019	0.0718	0.0275
Image segment	0.4455	0.0642	**0.4609**	0	**0.4609**	0	0.0133	0.0111
Cardiotocography	**0.0458**	0.0068	0.0263	0	0.0254	0.0043	0.0295	0.003
Student evaluation	**0.0569**	0.015	0.0511	0.0037	0.0538	0.0084	0.1321	0.063
Landsat satllite	0.065	0.0034	0.0521	0.0029	0.0575	0.0038	**0.0668**	0.0021
Pen-based digits	0.0408	0.0028	0.0331	0.0014	0.036	0.0011	**0.0416**	0.0022
Balance scale	**0.1579**	9e-04	0.1551	0.0033	0.1542	0.0026	0.1497	0.0044
Diabetes	**0.09**	0.0024	0.0895	0.0023	0.0874	0.0022	0.0846	0.009
Heart-statlog	0.1165	0.0067	**0.1235**	0.0136	0.1173	0.007	0.1191	0.0091
Ionosphere	0.1924	0.0364	0.3705	0.0786	0.2325	0.0316	**0.3933**	6e-04
Sonar	0.3698	0.0255	0.426	0.0619	0.3295	0.0139	**0.5015**	0
Vehicle	**0.1054**	0.0078	0.1041	0.0076	0.0912	0.0037	0.0898	0.0048
Waveform-5000	**0.3384**	0.0302	0.2584	0.0064	0.2669	0.0094	0.3105	0.0122

TABLE 8. Number of evaluations when objective function is to maximize DI [10].

Problem	Proposed		Gen-SA		DE		GA	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Iris	826	741	969	735	**742**	675	2248	3703
Glass	**2842**	3123	6246	1454	14054	6358	33366	20686
Ecoli	**6497**	3890	24037	0	17887	6072	24686	24976
Banknote authentication	**397**	404	524	522	478	563	2919	2313
Image segment	1427	942	2785	777	**803**	1635	11468	5243
Cardiotocography	2970	560	**1734**	0	3165	2737	9482	5068
Student evaluation	**113**	92	474	286	543	305	1067	730
Landsat satllite	352	62	**310**	250	1086	650	1078	666
Pen-based digits	258	26	**133**	86	789	441	859	358
Balance scale	**1040**	715	4683	1256	3761	1387	5883	4999
Diabetes	**2977**	1785	8571	1863	7100	2724	10375	9903
Heart-statlog	**4720**	2666	14126	3561	12925	4424	15649	16292
Ionosphere	**11535**	7410	35144	8361	34731	12267	51808	34172
Sonar	**16680**	13511	52235	15023	64795	18216	20737	20683
Vehicle	**9401**	6315	35856	7228	32749	7290	33799	30102
Waveform-5000	471	66	**424**	351	788	467	1668	767

proposed heuristic: (i) It has better solution quality (DI) than Gen-SA in seven problems; (ii) It has a solution quality (DI) equal to Gen-SA in four problems; (iii) It is better than DE in solution quality (DI) in ten problems; (iv) It is equal to DE in three problems; (v) It is better than GA in ten problems; and (vi) It is equal to GA in two problems.

TABLE 9. Comparison the DI [10] results of the proposed heuristic with others using t-tests.

Problem	vs. Gen-SA		vs. DE		vs. GA	
	p-value	remarks	p-value	remarks	p-value	remarks
Iris	0.002888	Worse	0.002888	Worse	2.531e-14	Better
Glass	0.3216	Equal	0.008797	Better	4.879e-37	Better
Ecoli	1.888e-08	Better	1.367e-31	Better	3.832e-24	Better
Banknote authentication	0.0003501	Worse	0.0002555	Worse	1.58e-07	Better
Image segment	0.09474	Equal	0.09474	Equal	3.438e-44	Better
Cardiotocography	2.718e-25	Better	5.655e-30	Better	1.011e-23	Better
Student evaluation	0.03753	Better	0.3078	Equal	9.417e-08	Worse
Landsat satllite	6.478e-33	Better	7.162e-16	Better	0.003038	Worse
Pen-based digits	2.402e-14	Better	4.46e-09	Better	0.2794	Equal
Balance scale	3.227e-07	Better	1.473e-13	Better	3.938e-18	Better
Diabetes	0.298	Equal	1.681e-07	Better	0.0001302	Better
Heart-statlog	0.001656	Worse	0.576	Equal	0.1028	Equal
Ionosphere	1.056e-22	Worse	5.971e-08	Worse	1.329e-38	Worse
Sonar	1.216e-07	Worse	3.961e-15	Better	3.445e-37	Worse
Vehicle	0.4126	Equal	6.766e-18	Better	1.06e-19	Better
Waveform-5000	9.921e-25	Better	4.802e-23	Better	7.514e-08	Better
Note: significance level = 0.05						

TABLE 10. Comparison the number of evaluations of the proposed heuristic with others using t-tests when objective function is DI [10].

Problem	vs. Gen-SA		vs. DE		vs. GA	
	p-value	remarks	p-value	remarks	p-value	remarks
Iris	0.3351	Equal	0.5562	Equal	0.01024	Better
Glass	1.374e-09	Better	2.308e-17	Better	4.169e-14	Better
Ecoli	1.978e-34	Better	3.041e-18	Better	5.178e-06	Better
Banknote authentication	0.1792	Equal	0.4123	Equal	5.624e-10	Better
Image segment	6.001e-12	Better	0.02182	Worse	1.995e-18	Better
Cardiotocography	1.225e-20	Worse	0.6238	Equal	4.282e-12	Better
Student evaluation	3.292e-08	Better	2.574e-09	Better	1.477e-08	Better
Landsat satllite	0.2792	Equal	1.636e-09	Better	3.944e-09	Better
Pen-based digits	1.586e-07	Worse	3.18e-06	Better	1.284e-08	Better
Balance scale	3.457e-29	Better	1.478e-19	Better	1.203e-08	Better
Diabetes	9.31e-28	Better	7.078e-14	Better	3.406e-06	Better
Heart-statlog	3.234e-26	Better	3.757e-18	Better	2.103e-05	Better
Ionosphere	7.347e-27	Better	1.443e-18	Better	6.086e-11	Better
Sonar	8.358e-22	Better	2.904e-26	Better	0.2488	Equal
Vehicle	3.482e-35	Better	6.236e-31	Better	7.411e-07	Better
Waveform-5000	0.3563	Equal	1.743e-05	Better	6.751e-15	Better
Note: significance level = 0.05						

TABLE 11. Summary of the comparisons using t-tests when objective function is DI [8].

Problem	vs. Gen-SA		vs. DE		vs. GA	
	CHI	Eval. count	CHI	Eval. count	CHI	Eval. count
Iris	-	=	-	=	+	+
Glass	=	+	+	+	+	+
Ecoli	+	+	+	+	+	+
Banknote authentication	-	=	-	=	+	+
Image segment	=	+	=	-	+	+
Cardiotocography	+	-	+	=	+	+
Student evaluation	+	+	=	+	-	+
Landsat satellite	+	=	+	+	-	+
Pen-based digits	+	-	+	+	=	+
Balance scale	+	+	+	+	+	+
Diabetes	=	+	=	+	+	+
Heart-statlog	-	+	=	+	=	+
Ionosphere	-	+	-	+	-	+
Sonar	-	+	+	+	-	=
Vehicle	=	+	+	+	+	+
Waveform-5000	+	=	+	+	+	+

The results in Table 10 suggests that the number of evaluations of the proposed heuristic are better or equal to that of the other algorithms (Gen-SA, DE and GA) in most of the problems.

Table 11 shows a summary of the comparisons using t-tests. The summary reveal the following information about the comparison of the proposed heuristic with Gen-SA: (i) In five problems, the proposed heuristic is better in terms

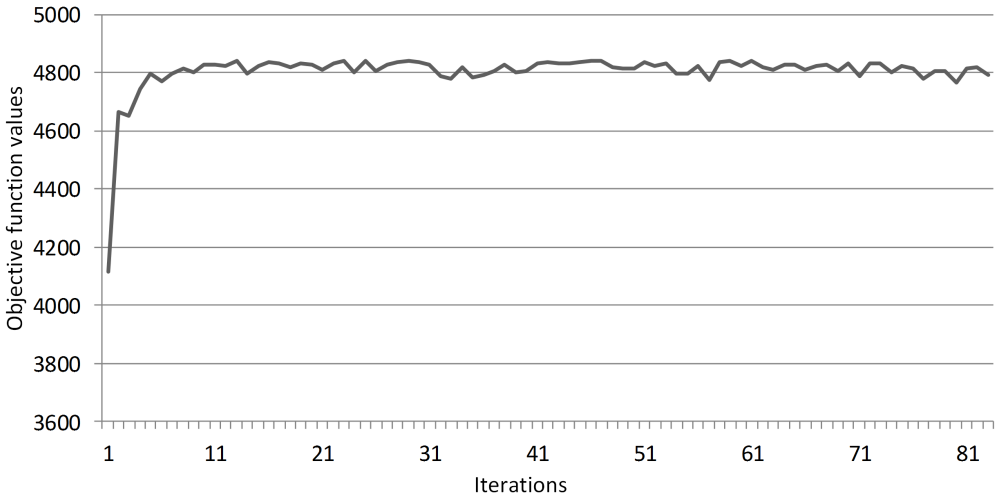


FIGURE 4. Optimization curve of the heuristic of the second part.

of solution quality (DI) and has number of evaluations equal or smaller than that of Gen-SA; (ii) In four problems, the proposed heuristic is equal to Gen-SA in solution quality (DI) but has better evaluation count; (iii) In two problems, the proposed heuristic has better solution quality but more number of evaluations; and (iii) In two problems, the Gen-SA has better solution quality and equal or smaller number of evaluations; and (iii) in three problems, the Gen-SA has better solution quality (DI) but has worser number of evaluations (since Gen-SA was allowed to execute for two-times more number of evaluations than the proposed heuristic). Table 11 also shows that the proposed heuristic is better than DE and GA in terms of both solution quality (DI) and number of evaluations in most of the problems.

Fig. 4 shows the objective function (CHI) versus iterations curve of the proposed optimization heuristic on the problem 'Landsat satellite'. The graph shows improvement in objective function value with iterations. The bad moves are also accepted in-order to increase diversity in the searching process and skip trapping in locally optimal values.

VI. CONCLUSION AND FUTURE WORK

This paper presented a heuristic for the clustering problem that can find centroids of clusters and uses fixed number of clusters. The proposed heuristic optimizes the compactness of clusters and separation between clusters. Two cluster validity indices CHI and DI have been used as objective functions in the proposed heuristic. The proposed heuristic consists of two parts. The first part is a greedy algorithm whose purpose is to find the data-points that can be used as centroids of clusters and maximize the separation between clusters. The second part is a heuristic that optimizes both compactness and separation. The heuristic comprises of some features of GA and SimE algorithms. In the proposed heuristic an attribute of a centroid is considered as a gene. The optimization heuristic determines values of the genes that yield clustering which is optimal w.r.t. a cluster validity index (CHI or DI). The proposed heuristic consists of five steps. First step is the initialization of the solution. Second step is the computation of fitness values of genes. Third step is the selection of a subset of genes that have low fitness values. Fourth step is the application of mutation operation on the selected genes. Fifth step is the acceptance or rejection of mutants based on the gain or loss in the value of the objective function. The performance of the proposed heuristic was evaluated on some real-life data-sets from the UCI repository and comparisons are performed with three other heuristics: Gen-SA, DE and GA. The Gen-SA and DE are standard Simulated Annealing and Differential Evolution algorithm and GA is a genetic algorithm proposed for the clustering problem. The experiments have two parts. The first part uses cluster validity index CHI as the objective function and the second part uses cluster validity index DI as the objective function. The experimental results show that the proposed heuristic can do efficient clustering w.r.t. a cluster validity index (CHI or DI) and requires fewer number of evaluations than the other heuristics.

ACKNOWLEDGEMENTS

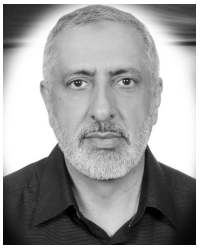
Acknowledgments are due to King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia for all support.

REFERENCES

[1] *Britannica Academic*, accessed on Jul. 8, 2017. [Online]. Available: http://academic.eb.com/levels/collegiate/article/605385
[2] C. C. Aggarwal and C. K. Reddy, *Data Clustering*. Boca Raton, FL, USA: Chapman & Hall, 2016.
[3] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1370–1386, Nov. 2004.
[4] P. Brucker, *On the Complexity of Clustering Problems*. Berlin, Germany: Springer, 1978, pp. 45–54. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-95322-4_5
[5] M. Greenacre and R. Primicerio, *Multivariate Analysis of Ecological Data*. Bilbao, Spain: Fundacin BBVA, 2013.

[6] P. A. Jaskowiak, R. J. G. B. Campello, and I. G. Costa, "Proximity measures for clustering gene expression microarray data: A validation methodology and a comparative analysis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 4, pp. 845–857, Jul. 2013.
[7] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intelli.*, vol. 1, no. 2, pp. 224–227, Apr. 1979.
[8] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.*, vol. 3, no. 1, pp. 1–27, Jan. 1974.
[9] J. C. Bezdek and N. R. Pal, "Cluster validation with generalized Dunn's indices," in *Proc. 2nd New Zealand Int. Two-Stream Conf. Artif. Neural Netw. Expert Syst.*, Dunedin, New Zealand, Nov. 1995, pp. 190–193.
[10] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *J. Cybern.*, vol. 4, no. 1, pp. 95–104, Jan. 1974.
[11] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.
[12] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, *Quality Scheme Assessment in the Clustering Process*. Berlin, Germany: Springer, 2000, pp. 265–276. [Online]. Available: http://dx.doi.org/10.1007/3-540-45372-5_26
[13] M. Nicholson, "Genetic algorithms and grouping problems," *Softw., Pract. Exper.*, vol. 28, no. 10, pp. 1137–1138, Aug. 1998. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1097-024X(199808)28:10%3C1137::AID-SPE192%3E3.0.CO;2-4/abstract
[14] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 2, pp. 133–155, Mar. 2009.
[15] S. M. Sait and H. Youssef, *Iterative Computer Algorithms With Applications in Engineering*. Los Alamitos, CA, USA: IEEE Computer Soc. Press, 1999.
[16] S. M. Sait and H. Youssef, *VLSI Physical Design Automation Theory and Practice*. Singapore: World Scientific, 1999.
[17] C. A. Suger and G. M. James, "Finding the number of clusters in a dataset: An information-theoretic approach," *J. Amer. Statist. Assoc.*, vol. 98, no. 463, pp. 750–763, Sep. 2003. [Online]. Available: https://search.proquest.com/docview/274839860?accountid=27795
[18] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 218–237, Jan. 2008.
[19] E. Cuevas, E. Santuario, D. Zaldivar, and M. Perez-Cisneros, "An improved evolutionary algorithm for reducing the number of function evaluations," *Intell. Autom. Soft Comput.*, vol. 22, no. 2, pp. 177–192, Apr. 2016.
[20] W. Zhu, Y. Tang, J.-A. Fang, and W. Zhang, "Adaptive population tuning scheme for differential evolution," *Inf. Sci.*, vol. 223, pp. 164–191, Feb. 2013. [Online]. Available:http://www.sciencedirect.com/science/article/pii/S0020025512006123
[21] M. S. Gibbs, H. R. Maier, and G. C. Dandy, "Using characteristics of the optimisation problem to determine the genetic algorithm population size when the number of evaluations is limited," *Environ. Model. Softw.*, vol. 69, pp. 226–239, Jul. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S1364815214002473
[22] Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng, "Generalized simulated annealing for global optimization: The gensa package," *R J.*, vol. 5, no. 1, pp. 13–29, Jun. 2013. [Online]. Available: http://journal.r-project.org/
[23] K. Mullen, D. Ardia, D. Gil, D. Windover, and J. Cline, "DEoptim: An R package for global optimization by differential evolution," *J. Statist. Softw.*, vol. 40, no. 6, pp. 1–26, Apr. 2011. [Online]. Available: http://www.jstatsoft.org/v40/i06/
[24] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognit.*, vol. 33, no. 9, pp. 1455–1465, Sep. 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320399001375
[25] *UCI Repository of Machine Learning Database*, (1998), accessed on Feb. 24, 2017. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLrepository.html
[26] *Software Environment for the Advancement of Scholarly Research (SEASR)*, (2008), accessed on Feb. 24, 2017. [Online]. Available: http://repository.seasr.org/Datasets/UCI/csv/
[27] S. Z. Selim and K. Alsultan, "A simulated annealing algorithm for the clustering problem," *Pattern Recognit.*, vol. 24, no. 10, pp. 1003–1008, Jan. 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ 0031320391900970
[28] Q. Kang, S. Liu, M. Zhou, and S. Li, "A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence," *Knowl.-Based Syst.*, vol. 104, pp. 156–164, 2016. [Online]. Available: http://www.sciencedirect. com/science/article/pii/S0950705116300739
[29] L. Pace, *Beginning R: An Introduction to Statistical Programming*. New York, NY, USA: Apress, 2012.
[30] A. Alajmi and J. Wright, "Selecting the most efficient genetic algorithm sets in solving unconstrained building optimization problem," *Int. J. Sustain. Built Environ.*, vol. 3, no. 1, pp. 18–26, Jun. 2014. [Online]. Available: http:// www.sciencedirect.com/science/article/pii/S2212609014000399

UMAIR F. SIDDIQI (M'12) was born in Karachi, Pakistan, in 1979. He received the B.E. degree in electrical engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 2002, the M.Sc. degree in computer engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2007, and the Dr. Eng. degree from Gunma University, Japan, in 2013. He is currently a Research Engineer with the Center of Communications and Information Technology Research of Research Institute, KFUPM. He has authored over 20 research papers in international journals and conferences. He also has two U.S. patents. His main areas of research interest are the application of evolutionary algorithms on engineering problems, electronic design automation, game theory, and different areas in artificial intelligence.



SADIQ M. SAIT (SM'02) was born in Bengaluru. He received the bachelor's degree in electronics engineering from Bangalore University in 1981, and the master's and Ph.D. degrees in electrical engineering from the King Fahd University of Petroleum and Minerals (KFUPM) in 1983 and 1987, respectively. He is currently a Professor of Computer Engineering and the Director of the Center for Communications and IT Research, Research Institute, KFUPM. He has authored over 200 research papers, contributed chapters to technical books, and lectured in over 25 countries. He is also the Principle Author of two books. He received the Best Electronic Engineer Award from the Indian Institute of Electrical Engineers, Bengaluru, in 1981.