Pertanika J. Sci. & Technol. 27 (1): 81 - 121 (2019)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Supervised Clustering based on a Multi-objective Genetic Algorithm

Vipa Thananant* and Surapong Auwatanamongkol

School of Applied Statistics, National Institute of Development Administration, Bangkok 10240, Thailand

ABSTRACT

Supervised clustering organizes data instances into clusters on the basis of similarities between the data instances as well as class labels for the data instances. Supervised clustering seeks to meet multiple objectives, such as compactness of clusters, homogeneity of data in clusters with respect to their class labels, and separateness of clusters. With these objectives in mind, a new supervised clustering algorithm based on a multi-objective crowding genetic algorithm, named SC-MOGA, is proposed in this paper. The algorithm searches for the optimal clustering solution that simultaneously achieves the three objectives mentioned above. The SC-MOGA performs very well on a small dataset, but for a large dataset it may not be able to converge to an optimal solution or can take a very long running time to converge to a solution. Hence, a data sampling method based on the Bisecting K-Means algorithm is also introduced, to find representatives for supervised clustering. This method groups the data instances of a dataset into small clusters, each containing data instances with the same class label. Data representatives are then randomly selected from each cluster. The experimental results show that SC-MOGA with the proposed data sampling method is very effective. It outperforms three previously proposed supervised clustering algorithms, namely SRIDHCR, LK-Means and SCEC, in terms of four cluster validity indexes. The experimental results show that the proposed data sampling method not only helps to reduce the number of data instances to be clustered by the SC-MOGA, but also enhances the quality of the data clustering results.

Keywords: Crowding genetic algorithm, data sampling, multi-objective optimization, Pareto optimal solutions, supervised clustering

ARTICLE INFO

Article history:
Received: 4 January 2018
Accepted: 27 September 2018
Published: 24 January 2019

E-mail addresses:
vipa@cpc.ac.th (Vipa Thananant)
surapong@as.nida.ac.th (Surapong Auwatanamongkol)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

Nowadays, very large amounts of data are generated and collected from diverse sources. There is a growing need to obtain useful information or patterns from data that have been collected. One of the essential tools for extracting such information or patterns is data clustering (Kaufman & Rousseeuw, 1990). Traditional (unsupervised) clustering tries to group data instances into clusters such that intra distances (distances between data instances in the same clusters) are minimal, while inter distances (distances between data instances from different clusters) are maximal (Jain & Dubes, 1988). Unsupervised clustering does not rely on predefined classes or class-labelled training examples like supervised clustering to group data instances into cluster. It is not necessarily guaranteed to group data objects with the same class together. Besides these two objectives, supervised clustering incorporates the third objective, which of minimal impurity level, which requires all data instances in each cluster to have the same class label. Some of the existing supervised clustering algorithms may consider different objectives – for example, SRIDHCR (Eick et al., 2004) considers only the impurity level and the number of clusters.

Supervised clustering is useful for various applications. In general, supervised clustering can be used for creating background knowledge for a dataset, dataset compression and editing (Eick et al., 2004), regional learning and evaluating distance functions in distance function learning (Eick et al., 2006). Finley and Joachims (2005) presented an SVM algorithm for training a clustering algorithm that optimized a variety of clustering performance measures. The algorithm had been used for noun-phrase and news article clustering. Eick et al. (2006) introduced a supervised clustering approach, SCAH algorithm, for region discovery. Haider et al. (2007) presented a supervised clustering algorithm for Streaming Data and applied it for email batch detection to filter spams. Maji (2010) proposed a novel supervised attribute clustering algorithm to find groups of co-regulated genes with respect to their gene expressions. Grbovic et al. (2013) studied supervised clustering, MM-PL algorithm, for the context of label ranking data. This algorithm can be used to divide the section of target marketing. Peralta et al. (2013) proposed LK-Means, an algorithm for supervised clustering based on a variant of K-Means which incorporated information about class labels. It had been shown that it could be used to generate a codebook for a visual recognition task.

Supervised Clustering problems can be viewed as optimization problems with multi objectives such as minimizing intra cluster distances, maximizing inter cluster distances and minimizing cluster impurity with respect to the class labels of data instances in clusters. To solve multi-objective optimization problems, evolutionary algorithms has become very popular due to their effectiveness to find the optimal solutions (Deb, 2001). An evolutionary algorithm based on a genetic algorithm had been proposed to solve supervised clustering problems. The algorithm, namely SCEC, combines multiple objective values of supervised

clustering into single objective value using a weighed sum of the multiple objective values. It has been shown to outperform some supervised clustering algorithms such as LK-means and SRIDHCR algorithm, using four evaluation metrics (Adjusted Mutual Information (AMI) (Vinh et al., 2009), Adjusted Variation of Information (AVI) (Vinh et al., 2009), Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) and Mirkin distance (MD) (Mirkin & Chernyj, 1970). These evaluation metrics are based on the contingency table shown in Table 1. Based on the results of the four evaluation metrics, they are still far from desirable values. SCEC also considers only two objectives for optimization, i.e. intra cluster distance (compactness) and cluster impurity. Moreover, the weighed sum scheme for combining multi objectives requires proper setting of the weight values which can be difficult for three or more objectives. It had been shown that the scheme cannot find the non-dominated or Pareto-optimal solutions if the Pareto-optimal front was non-convex (Deb, 2001).

Table 1
The contingency table

Y/Z	Z_1	Z_2	…	Z_B	Total
Y_1	u_{11}	u_{12}	…	u_{1B}	r_1
Y_2	u_{21}	u_{22}	…	u_{2B}	r_2
⋮	⋮	⋮	⋮	⋮	⋮
Y_A	u_{A1}	u_{A2}	…	u_{AB}	r_A
Total	c_1	c_2		c_B	N

Convergence and diversity are the two conflicting goals of evolutionary algorithm. On one hand, if the algorithm focuses more on the convergence to reach the optimal solutions, diversity of chromosomes in the population must be low so the search of the algorithm can be more focus on very good solutions in the population. This may lead to premature convergence to suboptimal solutions. On the other hand, if the algorithm focuses on diversity which allows the algorithm to search more broadly on potential solutions, the convergence becomes slow. A genetic algorithm faces the problem of trying to achieve the two conflicting goals at the same time. De Jong (1975) proposed crowding as a technique to improve population diversity in a genetic algorithm while maintaining the good convergence. The main concept of crowding is to replace a parent chromosome with its most similar offspring if it is fitter than the parent chromosome. With this replacement strategy, multiple subpopulations are formed which allow the search to continue in each population concurrently. This helps diversify the search to many parts of the search space and enhance the chance to find the optimal solutions. At the end, the crowding genetic

algorithm converges to multiple solutions, so it is suitable for multi-modal optimization problems.

In this paper, we propose a new supervised clustering algorithm based on a multi-objective crowding genetic algorithm. Unlike the SCEC, it considers three objective functions to optimize, i.e. intra cluster distances (compactness of clusters), inter cluster distances (separateness of clusters) and impurity levels of clusters. A Pareto ranking scheme is employed to rank chromosomes in the population based on the three objective functions. The ranks of the chromosomes are used in the parent replacement process of the crowding. The crowding can enhance the diversity of the search and so the chance of finding the optimal solutions. For a large dataset, the search space for the genetic algorithm can be very large which can prohibit the algorithm to converge to the optimal solutions. Therefore, a data sampling method based on clustering sampling approach is proposed to create a small set of data representatives for supervised clustering. A clustering algorithm based on bisecting K-means is used to group the data instances in the given dataset into clusters, each with data instances of the same class label. The data instances are then sampled from each cluster to form a representative dataset for clustering by the proposed algorithm. The results of experiments reveal that the proposed algorithm can find better clustering solutions than SCEC, SRIDHCR and LK-Means in terms of the four aforementioned evaluation metrics. The experimental results also show that the proposed sampling technique is effective to create a good representative dataset for the given dataset. The sampling technique not only helps reduce the running time of the proposed algorithm, but also helps the algorithm to converge better to the optimal solutions since it can reduce the size of search space for the genetic algorithm.

Supervised Clustering

Some existing supervised clustering algorithms are briefly presented in this section.

SRIDHCR

The objective of SRIDHCR is to minimize the following fitness function (f(X)):

$$f(X) = I(X) + \alpha * P(K) \qquad [1]$$

where X is a clustering solution containing K clusters, $I(X)$ is the average impurity level of the clusters (the average percentage of minority data instances in a cluster whose class labels are different from that of the majority), α is the weight (between 0 and 2) imposed on the penalty value $P(K)$ where P(K)=0 when K is less than the number of actual classes (A) in the given dataset and $P(K) = \sqrt{\frac{K-A}{N}}$ when K > = the number of actual classes, and N is the number of data instances.

For each run of SRIDHCR, a number of initial cluster representatives are selected randomly from the data instances. The remaining data instances are then assigned to their

closest representatives to form clusters. For each iteration, a new candidate set of cluster representatives is created by adding one data instance that is not a representative and removing one data instance that is. The current set of cluster representatives, *S,* is replaced with the best candidate set *X* for which $f(X)$ is better than $f(S)$. The iteration stops when no improvement of the fitness function value is achieved.This search process is tried for several runs and the best solution among all runs is reported. SRIDHCR also varies the value of *K* to determine its optimal value.

SCEC

SCEC (Eick et al., 2004) adopts the same objective function $f(X)$ as SRIDHCR. SCEC searches for the optimal set of cluster representatives by following an evolutionary computing approach. SCEC first randomly creates a population of solutions or chromosomes. Each of these specifies is a set of representatives of clusters. SCEC selects two chromosomes randomly from the population, and the chromosome with the better fitness value is chosen to become one of the parent chromosomes for the reproduction of offspring. Three genetic operators are used to create a new offspring chromosome for the next generation. These are crossover, mutation and copy operators. For the crossover, two parent chromosomes are recombined to create two offspring chromosomes. The mutation operator selects one data representative randomly and replaces it with a randomly selected non-representative one. The copy operator simply copies the parent chromosomes to the new population. Finally, to build a cluster, data instances are assigned to their nearest representatives. The evolutionary process is performed repeatedly until the population converges.

Labelled K-Means (LK-Means)

The LK-Means algorithm (Peralta et al., 2013) is similar to the traditional unsupervised K-Means algorithm, but class labels of the data instances are considered in the evaluation of the LK-Means fitness function. The fitness function of LK-Means is based on two criteria: (1) a discriminative score based on class labels; and (2) a generative score based on a traditional metric for unsupervised clustering. We assume a dataset X with N training instances (x_i , y_i), where $x_i \in R^d$, $y_i \in [1, \ldots, L]$, $i \in [1 \ldots N]$ and X is partitioned into K clusters. LK-Means replaces the traditional K-Means fitness function by the following Equation 2:

$$F(a_k^l, \partial_{nk}^l) = \sum_{n=1}^{N} \left[\beta \sum_{k=1}^{K} \sum_{l=1}^{L} \partial_{nk}^l \; \| x_n - a_k^l \|^2 \omega_k^l + (1-\beta) \sum_{k=1}^{K} \partial_{nk} \| x_n - a_k \|^2 \right] \quad [2]$$

where β and 1 - β are the weights for the supervised and unsupervised clustering scores, respectively. The value of β is between 0 and 1. a_k^l is the supervised mean of the data instances in cluster C_k with label l. ∂_{nk}^l is the supervised indicator that assigns instance x_n to the mean a_k^l . ω_k^l is a prior factor for data instances with label *l* inside cluster C_k. ∂_{nk}

is the unsupervised indicator for data instance x_n and cluster C_k. ∂_{nk} is the unsupervised mean for cluster C_k.

The LK-Means algorithm can be described as follows:

1. Initialize K initial means of clusters randomly.

2. Associate each data instance with the means of the clusters and compute the initial value of ∂_{nk}^{l} (using Equation 4 below)

3. Compute the supervised means a_k^l (using Equation 3) and then ω_k^l (using Equation 8)

4. Compute the unsupervised means a_k (using Equation 7)

5. Compute the supervised indicator ∂_{nk}^{l} (using Equation 5)

6. Compute the fitness function F (using Equation 2)

7. Repeat steps 3 to 6 until the value of the fitness function F converges (or the change of the fitness function value is below a given threshold).

To compute a_k^l use the following equation:

$$a_k^l = \frac{\beta \sum_{n=1}^{N} \partial_{nk}^{l} x_n + (1-\beta) \sum_{n=1}^{N} \partial_{nk} (x_n - \tilde{a}_k + \tilde{\omega}_k^l \tilde{a}_k^l)}{\beta \sum_{n=1}^{N} \partial_{nk}^{l} + (1-\beta) \tilde{\omega}_k^l \sum_{n=1}^{N} \partial_{nk}} \quad [3]$$

where $\tilde{a}_k$, $\tilde{\omega}_k^l$ and $\tilde{a}_k^l$ are the previous iteration values of a_k , ω_k^l and a_k^l

∂_{nk}^{l} , ∂_{nk} , $\tilde{a}_k$, $\tilde{\omega}_k^l$ and $\tilde{a}_k^l$ need to be initialized before computing a_k^l in Equation 3

The initial value of ∂_{nk}^{l} can be computed from:

$$\partial_{nk}^{l} = \frac{\theta_{nk}^{l} + \sigma}{1 + L * K * \sigma} \quad [4]$$

$$where \quad \theta_{nk}^{l} = \begin{cases} 1 & if\ x_n \in C_k \wedge y_n = l \\ 0 & otherwise \end{cases}$$

θ_{nk}^{l} is equal to one when the data instance x_n is in cluster C_k and has the class label of l. Otherwise it is equal to zero. A constant σ = 0.001 is the compensate value of the label uncertainty.

After initialization, ∂_{nk}^{l} can be evaluated from:

$$\partial_{nk}^{l} = \begin{cases} 1 & if\ k = argmin_j \left[\beta \partial_{nj}^{l} \left\| x_n - a_j^l \right\|^2 \omega_j^l + (1-\beta) \partial_{nj} \left\| x_n - a_j \right\|^2 \right] \\ 0 & otherwise \end{cases} \quad [5]$$

where ∂_{nk} can be computed using the following equation:

$$\partial_{nk} = \begin{cases} 1 & if\ x_n \in C_k \\ 0 & otherwise \end{cases} \quad [6]$$

a_k can be computed using the following equation:

$$a_k = \sum_{l=1}^{L} \omega_k^l \, a_k^l \quad [7]$$

ω_k^l using the following equation:

$$\omega_k^l = \frac{\sum_{n=1}^{N} \partial_{nk}^{l}}{\sum_{n=1}^{N} \partial_{nk}} \quad [8]$$

ω_k^l is between 0 and 1. When ω_k^l is equal to 1, all data instances in cluster k have only label *l*. On the other hand, when ω_k^l is equal to 0, no data instance in cluster k has label *l*.

Proposed Algorithm

SC-MOGA

The proposed algorithm searches for clustering solutions that minimize two objective functions, namely the impurity level (f_1) and the sum squared error (SSE) or compactness (f_2), and maximize the third objective function, namely the inter cluster distance or separateness (f_3), as follows:

$$f_1 = \sum_{i=1}^{K} the\ percentage\ of\ minority\ data\ instances\ in\ the\ i^{th}\ cluster$$

$$f_2 = \sum_{j=1}^{N} (Euclidean\ distance\ (x_j, the\ center\ of\ the\ cluster\ containing\ x_j))^2$$

$$f_3 = \frac{2}{K(K-1)} \sum_{s=1}^{K-1} \sum_{t=s+1}^{K} Euclidean\ distance(\ the\ center\ of\ cluster\ s\ , the\ center\ of\ cluster\ t)$$

where N is the number of data instances to be clustered and K is the number of clusters. SC-MOGA represents clustering solutions or chromosomes by integer encoding (Hruschka et al., 2009). Each gene in the chromosome is an integer between 1 and K. It represents

Algorithm SC–MOGA

```
1. n = 1
2. Initialize the chromosome population of size N randomly
while n <= the number of generations do
    3. The chromosomes in the current population are randomly paired
    4. Evaluate the three fitness function values of each chromosome and rank all
    chromosomes based on Pareto dominances
    for each pair of chromosomes do
        5. Recombine the two chromosomes, parent1 and parent2, to create
        two offspring, child1 and child2
        6. Mutate child1 and child2 with a predefined mutation probability
        7. Rank child1 and child2 against the current population
        if    distance(parent1 , child1) + distance(parent2 , child2) <
              distance(parent1 , child2) + distance(parent2 , child1) then
              q1 = either parent1 or child1 whichever has better ranking
              q2 = either parent2 or child2 whichever has better ranking
        else
              q1 = either parent1 or child2 whichever has better ranking
              q2 = either parent2 or child1 whichever has better ranking
        end if
        8. Place q1 and q2 in the new population
    end for
    9. Replace the current population with the new population
    10. n = n + 1
```

the identity of the cluster to which the corresponding data instance is assigned (Figure 1). The length of the chromosome is therefore equal to the number of data instances.

With this encoding scheme, the shape of each cluster defined by a chromosome can be globular or non-globular one. However, one distinct clustering can have several chromosome representations. For instance, the chromosomes [1,1,1,1,2,2,2,2,3,3], [1,1,1,1,3,3,3,3,2,2] and [2,2,2,2,1,1,1,1,3,3] represent the same clustering solution: the clustering contains three clusters, the first cluster with the first four data instances, the second with the next four data instances and the third with the last two data instances. Finding clustering solutions that optimize the three objective functions therefore becomes a multi-modal multi-objective optimization problem, since the same optimal clustering can be represented by multiple solutions in the search space. A multi-objective crowding genetic algorithm method is chosen as the search method for SC-MOGA, since it can converge to multiple solutions simultaneously. The SC-MOGA algorithm is summarized below:

The solution with the best ranking in the final population becomes the clustering solution. When there is a tie on ranking, the orders of the solutions are considered to break the tie.

In Steps 4 and 7: Ranking a chromosome is performed against the current population, based on how many chromosomes there are in the population that are dominated by the chromosome (Fonseca & Fleming, 1993). Suppose a chromosome A has three fitness values f_{1A}, f_{2A}, f_{3A} and a chromosome B has three fitness values f_{1B}, f_{2B}, f_{3B}. The goal is to minimize the two fitness values f_1, f_2 and maximize the fitness value f_3, then chromosome A dominates chromosome B when $f_{1A} \leq f_{1B}$ and $f_{2A} \leq f_{2B}$ and $f_{3A} \geq f_{3B}$.

In Step 5: Uniform crossover (Syswerda, 1989) is applied with a crossover probability = 0.5. The uniform crossover is used instead of one point or two point crossover since it is not sensitive to the order of data instances of the chromosome encoding.

In Step 6: Mutation is performed with a given mutation probability on each gene in a chromosome. It assigns each gene a new random value between 1 and K.

In Step 7 and 8: distance (i_1, i_2) is the distance between two chromosomes i_1 and i_2. It is measured in terms of Hamming distance (Hamming, 1950). Based on the crowding method (De Jong, 1975), a parent chromosome will be replaced by its closest (most similar) child if the child is fitter than its parent (has higher Pareto ranking than its parent). Either the parent chromosome or its closest child with better ranking than the parent chromosome is kept temporarily in the new population.

In step 9 and10: the new population replaces the current population and the algorithm proceeds to the next generation.

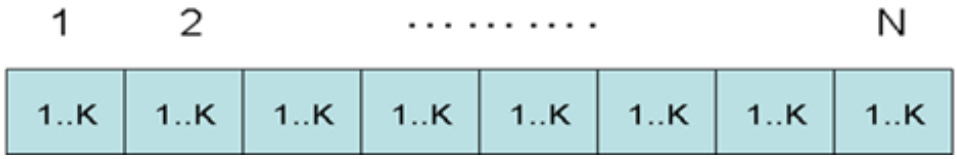


Figure 1. Chromosome Encoding for SC-MOGA

SC-MOGA with Data Sampling

For very large datasets, the search space for SC-MOGA also becomes very large, so it may not be able to converge to the optimal solution or can take a significant amount of running time before it converges. To overcome the problem, a stratified random sampling (Cadima et al., 2005) is performed on the given dataset to get a small representative dataset for SC-MOGA. The sampling is based on segmenting the dataset into compact clusters. Each cluster contains data with the same class label, the clusters represent strata and so a number of data instances are randomly sampled from each strata. To segment the dataset as mentioned above, a Bisecting K-means approach is adopted. The stratified random sampling procedure is summarized below:

procedure STRATIFIED_RANDOM_SAMPLING

1. Split the given dataset into two clusters using the K-Means algorithm (K = 2) and insert the two clusters into an empty list of candidate clusters for Bisecting
2. **repeat**
   Remove the cluster with the highest impurity level (represented by the percentage of minority data instances whose class labels are different from that of the majority) from the list and split the cluster into two using the K-Means algorithm. Then insert the clusters back into the list. When there is a tie on impurity, remove the cluster with the highest SSE

until all clusters in the list contain no impurity (zero impurity level)

3. Proportionally allocate n_h data samples to each cluster or stratum in the list of m clusters, as follows

$$n_h = S * \left(\frac{N_h\, SE_h}{\sum_{i=1}^{m} N_i\, SE_i}\right)$$

where n_h represents the sample size allocated to each stratum h, N_i represents the size of stratum i, SE_i represents the sum squared error of each stratum i as follows

$$SE_i = \sum_{j=1}^{N_i} (Euclidean\ distance(x_j, stratum\_center(i)))^2$$

x_j represents the j[th] data instance of stratum i and stratum_center(i) represents the center of stratum i, S represents the expected sample size calculated from (Krejcie & Morgan, 1970)

$$S = \frac{\chi^2\, NF(1-F)}{(d^2\,(N-1)) + (\chi^2\, F(1-F))}$$

where χ^2 is the designated chi-square value, N is the size of the population, F is the population fraction, and d is the precision degree. Each stratum is allocated at least one data sample ($n_i >= 1$). Hence the total sample size is:

$$Sample\ Size = \sum_{k=1}^{m} n_k$$

4. For each stratum i, randomly sample n_i data instances from the stratum. All samples now form a representative dataset for the SC-MOGA.

end procedure

After performing SC-MOGA on the representative dataset, representatives of each stratum may be grouped into different clusters. Since each stratum represents a compact group of data instances with the same class label, all data in each stratum should be assigned to the same SC-MOGA cluster. The winner-take-all strategy is adopted in this case, so that all data instances in a stratum are assigned to the same cluster, the one that has the highest number of representatives of the stratum. If there is more than one such cluster, one can be chosen randomly among them.

EXPERIMENTS AND RESULTS

Two experiments were conducted to evaluate the performances of the proposed algorithms. A notebook with 2.5 GHz Core i5 processor and 4GB of RAM was used to run all the test cases in the experiments.

The First Experiment

The first experiment is intended to evaluate the performance of SC-MOGA and SC-MOGA with data sampling against some existing algorithms, namely LK-Means and SRIDHCR. Because of the difficulties we encountered when implementing the two algorithms, the experimental results for LK-Means and SRIDHCR on eight datasets in Peralta et al. (2013) were used for the comparison. The eight datasets, taken from the UCI Machine Learning Repository (Lichman, 2013), are Iris, Statlog ('Heart'), Glass Identification ('Glass'), Pima Indians Diabetes ('Diabetes'), Statlog ('Vehicle Silhouettes'), Image Segmentation, Ionosphere and Connectionist Bench (Sonar, Mines vs. Rocks) ('Sonar'). Before performing the clustering, the variable values in the datasets were normalized using a max-min normalization scheme. The details of the dataset are shown in Table 2.

Table 2
Details of the eight UCI Machine Learning Repository Datasets

Dataset Name	# objects	# variables	# classes
Iris	150	4	3
Heart	270	13	2
Glass	214	9	6
Diabetes	768	8	2
Vehicle Silhouettes	846	18	4
Image Segmentation	2310	19	7
Ionosphere	351	34	2
Sonar	208	60	2

To measure the performance of the SC-MOGA, we applied ten-fold cross validation. For each fold, the genetic algorithm was tried for five runs. The five solutions, one from each run, were ranked, and the one with the highest Pareto ranking among them was selected as the optimal solution for the fold. Finally, the averages of the four metrics of the ten folds for the optimal solutions were computed. We followed the experiments of Peralta et al. (2013) that selected five values for the number of clusters (K) ranging from the lower bound value, which is the actual number of classes, to the upper bound value of $\lfloor \sqrt{number\ of\ data\ instances\ (or\ sample\ size)/2} \rfloor$ with equal intervals. Several trials of the experiment were conducted with varying parameter values. The best performance of the algorithm in terms of cluster validity indexes were achieved with the following setting of the parameter values:

Size of population = N log K
Crossover Operations = Uniform crossover with probability of 0.5
Mutation probability = 0.01
Number of generations per run = 500 (except that for Vehicle Silhouettes the number was 1,000 and for Image Segmentation it was 10,000)

Table 3

Sampling parameter values used by SC-MOGA with data sampling on the first eight datasets

Dataset	Number of Data Objects (N)	χ^2 chi-square	Population Fraction (F)	Precision Degree (d)	Expected Sample Size (S)	Number of Strata (m)	Actual sample size $\sum_{k=1}^{m} n_k$.
Iris	150	3.841	0.5	0.05	109	14	109
Heart	270	3.841	0.5	0.05	159	191	260
Glass	214	3.841	0.5	0.05	138	22	138
Diabetes	768	3.841	0.5	0.05	257	529	605
Vehicle Silhouettes	846	3.841	0.5	0.05	265	749	818
Image Segmentation	2310	3.841	0.5	0.05	330	1105	1191
Ionosphere	351	3.841	0.5	0.05	184	240	335
Sonar	208	3.841	0.5	0.05	136	8	136

Table 3 shows all the sampling parameter values used in the experiments for SC-MOGA with data sampling. The four cluster validity indexes were used for the performance comparison. The validity index results for the eight datasets are shown in Tables 4 to 7 (the first eight datasets in each of these tables). The results of SCEC and the rest of datasets in the tables come from the second experiment and will be explained later. One-sided paired t-tests were performed to compare the performances of SC-MOGA, SC-MOGA with data sampling, LK-Means and SRIDHCR. To do the t-test between two algorithms for

each index, we computed the differences between the indexes achieved by the two algorithms for the eight datasets and for all Ks (number of clusters). The results of the tests are shown in Tables 8 and 9. The results in Table 8 show that SC-MOGA achieved better performances than SRIDHCR and LK-Means, with a confidence of more than 95% for all four indexes, but its performance was not better than SC-MOGA with data sampling. The results in Table 9 show that SC-MOGA with data sampling achieved a better performance than SRIDHCR, LK-Means and SC-MOGA with a confidence level of more than 95% for all four indexes.

Table 4

Adjusted Mutual Information (AMI) results on 23 datasets

Dataset			**Number of clusters**			
Iris	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	0.196	0.260	0.204	0.236	0.241	0.227
LK-Means	0.655	0.538	0.497	0.451	0.387	0.506
SCEC	0.912	0.691	0.589	0.543	0.483	0.644
SC-MOGA	0.912	0.740	0.611	0.548	0.503	0.663
SC-MOGA with data sampling	1	1	0.950	0.908	0.908	0.953
Heart	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.011	0.082	0.078	0.104	0.097	0.074
LK-Means	0.293	0.212	0.137	0.134	0.104	0.176
SCEC	0.417	0.247	0.185	0.192	0.171	0.242
SC-MOGA	0.658	0.441	0.331	0.295	0.259	0.397
SC-MOGA with data sampling	0.719	0.441	0.350	0.295	0.272	0.415
Glass	**K=6**	**K=7**	**K=8**	**K=9**	**K=10**	**Mean**
SRIDHCR	0.093	0.132	0.138	0.092	0.106	0.112
LK-Means	0.148	0.159	0.156	0.132	0.149	0.149
SCEC	0.478	0.492	0.457	0.437	0.432	0.459
SC-MOGA	0.532	0.577	0.516	0.482	0.467	0.515
SC-MOGA with data sampling	0.759	0.718	0.743	0.715	0.715	0.730
Diabetes	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	0.113	0.049	0.044	0.043	0.041	0.058
LK-Means	0.086	0.068	0.047	0.040	0.043	0.057
SCEC	0.194	0.089	0.095	0.080	0.081	0.108
SC-MOGA	0.375	0.269	0.198	0.160	0.129	0.226
SC-MOGA with data sampling	0.430	0.280	0.234	0.188	0.141	0.255

Table 4 *(Continue)*

Dataset	**Number of clusters**					
Vehicle Silhouettes	**K=4**	**K=8**	**K=12**	**K=16**	**K=20**	**Mean**
SRIDHCR	0.076	0.107	0.132	0.141	0.116	0.114
LK-Means	0.112	0.129	0.128	0.118	0.117	0.121
SCEC	0.270	0.325	0.297	0.268	0.207	0.273
SC-MOGA	0.492	0.417	0.345	0.369	0.335	0.392
SC-MOGA with data sampling	0.522	0.429	0.393	0.380	0.346	0.414
Image Segmentation	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	0.446	0.522	0.469	0.428	0.392	0.451
LK-Means	0.548	0.551	0.492	0.439	0.411	0.488
SCEC	0.755	0.655	0.600	0.540	0.465	0.603
SC-MOGA	0.625	0.601	0.565	0.521	0.465	0.555
SC-MOGA with data sampling	0.750	0.653	0.592	0.556	0.486	0.607
Ionosphere	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.053	0.112	0.069	0.082	0.075	0.078
LK-Means	0.174	0.177	0.125	0.156	0.108	0.148
SCEC	0.371	0.304	0.299	0.281	0.248	0.301
SC-MOGA	0.390	0.375	0.301	0.273	0.236	0.315
SC-MOGA with data sampling	0.511	0.417	0.361	0.313	0.284	0.377
Sonar	**K=2**	**K=4**	**K=6**	**K=8**	**K=10**	**Mean**
SRIDHCR	0.012	0.001	0.050	0.019	0.020	0.020
LK-Means	0.094	0.036	0.017	0.039	0.058	0.049
SCEC	0.225	0.198	0.180	0.192	0.191	0.197
SC-MOGA	0.474	0.484	0.414	0.349	0.311	0.406
SC-MOGA with data sampling	1	0.961	0.859	0.961	0.961	0.948
BS	**K=3**	**K=7**	**K=11**	**K=15**	**K=19**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.508	0.301	0.256	0.246	0.227	0.308
SC-MOGA	0.608	0.342	0.275	0.238	0.214	0.335
SC-MOGA with data sampling	0.613	0.364	0.292	0.257	0.235	0.352

Table 4 *(Continue)*

Dataset	**Number of clusters**					
BTSC	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.094	0.094	0.038	0.045	0.038	0.062
SC-MOGA	0.147	0.169	0.143	0.098	0.038	0.119
SC-MOGA with data sampling	0.292	0.334	0.280	0.233	0.225	0.273
CMSC	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.005	0.009	0.031	0.037	0.050	0.026
SC-MOGA	0.035	0.032	0.036	0.035	0.036	0.035
SC-MOGA with data sampling	0.473	0.311	0.268	0.254	0.198	0.301
CMC	**K=3**	**K=9**	**K=15**	**K=21**	**K=27**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.054	0.057	0.049	0.044	0.043	0.049
SC-MOGA	0.228	0.092	0.074	0.009	0.007	0.082
SC-MOGA with data sampling	0.650	0.801	0.566	0.401	0.396	0.563
HS	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.099	0.059	0.062	0.055	0.051	0.065
SC-MOGA	0.118	0.285	0.237	0.244	0.222	0.221
SC-MOGA with data sampling	0.152	0.355	0.373	0.329	0.320	0.306
LD	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.048	0.074	0.065	0.076	0.081	0.069
SC-MOGA	0.255	0.451	0.338	0.285	0.257	0.317
SC-MOGA with data sampling	0.351	0.496	0.597	0.637	0.530	0.522

Table 4 *(Continue)*

Dataset	**Number of clusters**					
MP	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.310	0.301	0.274	0.233	0.247	0.273
SC-MOGA	0.497	0.390	0.294	0.253	0.205	0.328
SC-MOGA with data sampling	0.616	0.431	0.369	0.341	0.296	0.411
Musk	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.148	0.120	0.117	0.143	0.117	0.129
SC-MOGA	0.255	0.312	0.303	0.257	0.200	0.265
SC-MOGA with data sampling	0.889	0.790	0.981	0.881	0.915	0.891
Seeds	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.777	0.669	0.669	0.541	0.480	0.627
SC-MOGA	0.828	0.697	0.591	0.539	0.471	0.625
SC-MOGA with data sampling	1	1	0.767	0.675	0.644	0.817
SPECTF	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.050	0.092	0.086	0.085	0.098	0.082
SC-MOGA	0.061	0.101	0.260	0.230	0.205	0.171
SC-MOGA with data sampling	1	1	1	1	1	1
SPF	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.385	0.394	0.299	0.277	0.275	0.326
SC-MOGA	0.227	0.150	0.064	0.006	0.007	0.091
SC-MOGA with data sampling	0.574	0.707	0.694	0.599	0.607	0.636

Table 4 *(Continue)*

Dataset	**Number of clusters**					
TAE	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.141	0.122	0.114	0.134	0.116	0.125
SC-MOGA	0.432	0.465	0.369	0.461	0.411	0.428
SC-MOGA with data sampling	0.649	0.638	0.534	0.566	0.503	0.578
Vertebral	**K=3**	**K=6**	**K=9**	**K=12**	**K=15**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.548	0.548	0.338	0.321	0.281	0.407
SC-MOGA	0.557	0.548	0.431	0.420	0.326	0.456
SC-MOGA with data sampling	0.667	0.666	0.578	0.618	0.652	0.636
Wilt	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.019	0.021	0.024	0.026	0.010	0.020
SC-MOGA	0.027	0.049	0.055	0.059	0.040	0.046
SC-MOGA with data sampling	0.879	0.800	1	0.853	0.793	0.865
Wine	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.925	0.693	0.629	0.528	0.461	0.647
SC-MOGA	0.972	0.713	0.604	0.528	0.471	0.658
SC-MOGA with data sampling	1	0.902	0.939	0.748	0.738	0.865

Table 5

Adjusted Rand Index (ARI) results on 23 datasets

Dataset	**Number of clusters**					
Iris	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	0.190	0.259	0.196	0.221	0.233	0.220
LK-Means	0.644	0.568	0.552	0.527	0.457	0.550
SCEC	0.922	0.740	0.627	0.571	0.471	0.666
SC-MOGA	0.922	0.792	0.656	0.532	0.504	0.681
SC-MOGA with data sampling	1	1	0.980	0.961	0.961	0.980
Heart	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.019	0.110	0.099	0.111	0.099	0.088
LK-Means	0.315	0.257	0.164	0.155	0.119	0.202
SCEC	0.525	0.368	0.193	0.227	0.164	0.295
SC-MOGA	0.763	0.436	0.254	0.211	0.150	0.363
SC-MOGA with data sampling	0.816	0.439	0.306	0.212	0.184	0.391
Glass	**K=6**	**K=7**	**K=8**	**K=9**	**K=10**	**Mean**
SRIDHCR	0.074	0.092	0.104	0.079	0.091	0.088
LK-Means	0.168	0.134	0.143	0.119	0.137	0.140
SCEC	0.413	0.392	0.419	0.355	0.364	0.389
SC-MOGA	0.424	0.460	0.395	0.388	0.336	0.401
SC-MOGA with data sampling	0.786	0.776	0.754	0.732	0.666	0.743
Diabetes	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	0.182	0.059	0.068	0.048	0.041	0.080
LK-Means	0.150	0.089	0.060	0.043	0.045	0.077
SCEC	0.292	0.115	0.116	0.058	0.057	0.128
SC-MOGA	0.370	0.269	0.140	0.083	0.069	0.186
SC-MOGA with data sampling	0.508	0.284	0.146	0.094	0.062	0.219
Vehicle Silhouettes	**K=4**	**K=8**	**K=12**	**K=16**	**K=20**	**Mean**
SRIDHCR	0.051	0.082	0.109	0.110	0.088	0.088
LK-Means	0.082	0.103	0.108	0.098	0.100	0.098
SCEC	0.260	0.295	0.243	0.216	0.185	0.240
SC-MOGA	0.465	0.378	0.269	0.273	0.219	0.321
SC-MOGA with data sampling	0.506	0.385	0.318	0.264	0.222	0.339

Table 5 *(Continue)*

Dataset	**Number of clusters**					
Image Segmentation	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	0.446	0.483	0.410	0.326	0.278	0.389
LK-Means	0.447	0.502	0.444	0.388	0.357	0.428
SCEC	0.716	0.621	0.585	0.521	0.442	0.577
SC-MOGA	0.532	0.517	0.480	0.457	0.397	0.477
SC-MOGA with data sampling	0.692	0.589	0.502	0.496	0.409	0.538
Ionosphere	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.115	0.163	0.112	0.120	0.080	0.118
LK-Means	0.196	0.199	0.130	0.189	0.124	0.168
SCEC	0.447	0.426	0.408	0.366	0.247	0.379
SC-MOGA	0.387	0.390	0.301	0.279	0.248	0.321
SC-MOGA with data sampling	0.630	0.499	0.364	0.292	0.264	0.410
Sonar	**K=2**	**K=4**	**K=6**	**K=8**	**K=10**	**Mean**
SRIDHCR	0.042	0.001	0.048	0.018	0.025	0.027
LK-Means	0.103	0.044	0.034	0.052	0.059	0.058
SCEC	0.297	0.278	0.222	0.202	0.174	0.235
SC-MOGA	0.575	0.556	0.417	0.305	0.242	0.419
SC-MOGA with data sampling	1	0.990	0.941	0.990	0.990	0.982
BS	**K=3**	**K=7**	**K=11**	**K=15**	**K=19**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.594	0.278	0.192	0.156	0.124	0.269
SC-MOGA	0.649	0.273	0.186	0.127	0.100	0.267
SC-MOGA with data sampling	0.649	0.305	0.192	0.157	0.128	0.286
BTSC	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.227	0.227	0.027	0.038	0.032	0.110
SC-MOGA	0.066	0.116	0.076	0.032	0.013	0.061
SC-MOGA with data sampling	0.470	0.508	0.493	0.474	0.492	0.487

Table 5 *(Continue)*

Dataset	**Number of clusters**					
CMSC	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.005	0.005	0.007	0.014	0.017	0.010
SC-MOGA	0.086	0.092	0.069	0.014	0.017	0.056
SC-MOGA with data sampling	0.679	0.509	0.476	0.462	0.432	0.512
CMC	**K=3**	**K=9**	**K=15**	**K=21**	**K=27**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.055	0.043	0.033	0.025	0.021	0.035
SC-MOGA	0.225	0.078	0.053	0.005	0.004	0.073
SC-MOGA with data sampling	0.619	0.904	0.658	0.396	0.416	0.599
HS	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.229	0.106	0.088	0.056	0.037	0.103
SC-MOGA	0.083	0.245	0.161	0.136	0.113	0.148
SC-MOGA with data sampling	0.292	0.565	0.616	0.538	0.591	0.520
LD	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.085	0.092	0.078	0.060	0.074	0.078
SC-MOGA	0.331	0.474	0.273	0.194	0.153	0.285
SC-MOGA with data sampling	0.379	0.713	0.818	0.853	0.744	0.701
MP	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.249	0.377	0.191	0.138	0.141	0.219
SC-MOGA	0.514	0.349	0.199	0.144	0.102	0.262
SC-MOGA with data sampling	0.648	0.622	0.508	0.505	0.433	0.543

Table 5 *(Continue)*

Dataset	**Number of clusters**					
Musk	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.182	0.094	0.092	0.111	0.092	0.114
SC-MOGA	0.278	0.306	0.218	0.147	0.100	0.210
SC-MOGA with data sampling	0.934	0.923	0.996	0.967	0.977	0.959
Seeds	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.824	0.772	0.772	0.632	0.480	0.696
SC-MOGA	0.846	0.744	0.572	0.529	0.383	0.615
SC-MOGA with data sampling	1	1	0.834	0.722	0.710	0.853
SPECTF	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.141	0.077	0.053	0.044	0.043	0.072
SC-MOGA	0.143	0.102	0.170	0.143	0.109	0.133
SC-MOGA with data sampling	1	1	1	1	1	1
SPF	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.301	0.304	0.268	0.201	0.178	0.250
SC-MOGA	0.223	0.120	0.041	0.003	0.003	0.078
SC-MOGA with data sampling	0.533	0.702	0.710	0.594	0.600	0.628
TAE	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.147	0.150	0.121	0.124	0.105	0.129
SC-MOGA	0.437	0.486	0.299	0.455	0.369	0.409
SC-MOGA with data sampling	0.650	0.610	0.399	0.619	0.523	0.560

Table 5 *(Continue)*

Dataset			**Number of clusters**			
Vertebral	**K=3**	**K=6**	**K=9**	**K=12**	**K=15**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.646	0.646	0.316	0.313	0.236	0.431
SC-MOGA	0.447	0.535	0.360	0.327	0.219	0.378
SC-MOGA with data sampling	0.730	0.661	0.453	0.649	0.666	0.632
Wilt	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.002	0.005	0.002	0.001	0.001	0.002
SC-MOGA	0.018	0.010	0.009	0.007	0.006	0.010
SC-MOGA with data sampling	0.956	0.926	1	0.941	0.911	0.947
Wine	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.947	0.732	0.659	0.509	0.381	0.646
SC-MOGA	0.982	0.744	0.630	0.499	0.421	0.655
SC-MOGA with data sampling	1	0.944	0.971	0.797	0.831	0.909

Table 6

Adjusted Variation of Information (AVI) results on 23 datasets

Dataset			**Number of clusters**			
Iris	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	0.224	0.286	0.221	0.261	0.280	0.254
LK-Means	0.715	0.613	0.591	0.560	0.485	0.593
SCEC	0.913	0.799	0.727	0.698	0.646	0.757
SC-MOGA	0.913	0.850	0.750	0.708	0.669	0.778
SC-MOGA with data sampling	1	1	0.975	0.952	0.952	0.976

Table 6 *(Continue)*

Dataset	**Number of clusters**					
Heart	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.011	0.112	0.115	0.163	0.159	0.112
LK-Means	0.300	0.271	0.188	0.201	0.164	0.225
SCEC	0.420	0.319	0.273	0.289	0.267	0.314
SC-MOGA	0.660	0.612	0.498	0.456	0.411	0.527
SC-MOGA with data sampling	0.720	0.612	0.518	0.456	0.428	0.547
Glass	**K=6**	**K=7**	**K=8**	**K=9**	**K=10**	**Mean**
SRIDHCR	0.100	0.149	0.150	0.109	0.128	0.127
LK-Means	0.216	0.183	0.187	0.154	0.179	0.184
SCEC	0.487	0.499	0.486	0.497	0.508	0.495
SC-MOGA	0.550	0.615	0.580	0.550	0.546	0.568
SC-MOGA with data sampling	0.823	0.808	0.781	0.779	0.733	0.785
Diabetes	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	0.117	0.071	0.068	0.068	0.068	0.078
LK-Means	0.105	0.092	0.069	0.065	0.069	0.080
SCEC	0.194	0.131	0.146	0.130	0.132	0.147
SC-MOGA	0.388	0.398	0.313	0.260	0.213	0.314
SC-MOGA with data sampling	0.444	0.413	0.370	0.307	0.233	0.353
Vehicle Silhouettes	**K=4**	**K=8**	**K=12**	**K=16**	**K=20**	**Mean**
SRIDHCR	0.076	0.107	0.132	0.141	0.116	0.114
LK-Means	0.121	0.149	0.164	0.157	0.163	0.151
SCEC	0.276	0.383	0.378	0.350	0.291	0.336
SC-MOGA	0.500	0.493	0.441	0.491	0.458	0.477
SC-MOGA with data sampling	0.529	0.512	0.502	0.506	0.473	0.504
Image Segmentation	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	0.561	0.583	0.568	0.544	0.513	0.554
LK-Means	0.570	0.615	0.580	0.549	0.531	0.569
SCEC	0.758	0.740	0.730	0.720	0.704	0.730
SC-MOGA	0.640	0.619	0.585	0.556	0.501	0.580
SC-MOGA with data sampling	0.778	0.746	0.735	0.724	0.671	0.731

Table 6 *(Continue)*

Dataset	**Number of clusters**					
Ionosphere	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.059	0.132	0.084	0.106	0.106	0.097
LK-Means	0.182	0.225	0.173	0.212	0.160	0.190
SCEC	0.418	0.400	0.420	0.409	0.381	0.406
SC-MOGA	0.402	0.523	0.453	0.429	0.389	0.439
SC-MOGA with data sampling	0.520	0.554	0.531	0.477	0.441	0.505
Sonar	**K=2**	**K=4**	**K=6**	**K=8**	**K=10**	**Mean**
SRIDHCR	0.012	0.001	0.065	0.028	0.033	0.028
LK-Means	0.099	0.048	0.026	0.056	0.075	0.061
SCEC	0.225	0.256	0.255	0.281	0.285	0.260
SC-MOGA	0.478	0.626	0.586	0.518	0.475	0.537
SC-MOGA with data sampling	1	0.980	0.924	0.980	0.980	0.973
BS	**K=3**	**K=7**	**K=11**	**K=15**	**K=19**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.540	0.406	0.368	0.365	0.343	0.404
SC-MOGA	0.646	0.464	0.396	0.356	0.327	0.438
SC-MOGA with data sampling	0.650	0.486	0.420	0.379	0.356	0.458
BTSC	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.101	0.101	0.060	0.073	0.064	0.080
SC-MOGA	0.150	0.261	0.232	0.164	0.064	0.174
SC-MOGA with data sampling	0.299	0.464	0.401	0.344	0.335	0.369
CMSC	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.007	0.015	0.055	0.068	0.090	0.047
SC-MOGA	0.044	0.052	0.062	0.064	0.065	0.057
SC-MOGA with data sampling	0.546	0.457	0.405	0.392	0.314	0.423

Table 6 *(Continue)*

Dataset	**Number of clusters**					
CMC	**K=3**	**K=9**	**K=15**	**K=21**	**K=27**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.055	0.076	0.070	0.065	0.065	0.066
SC-MOGA	0.232	0.126	0.107	0.013	0.011	0.098
SC-MOGA with data sampling	0.432	0.875	0.696	0.546	0.532	0.616
HS	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.110	0.080	0.092	0.086	0.083	0.090
SC-MOGA	0.127	0.417	0.363	0.392	0.363	0.332
SC-MOGA with data sampling	0.157	0.473	0.512	0.466	0.458	0.413
LD	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.053	0.101	0.096	0.118	0.128	0.099
SC-MOGA	0.257	0.618	0.506	0.442	0.408	0.446
SC-MOGA with data sampling	0.353	0.664	0.717	0.753	0.669	0.631
MP	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.342	0.398	0.422	0.367	0.396	0.385
SC-MOGA	0.511	0.562	0.453	0.401	0.331	0.452
SC-MOGA with data sampling	0.625	0.572	0.514	0.487	0.434	0.526
Musk	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.149	0.184	0.180	0.224	0.180	0.183
SC-MOGA	0.256	0.438	0.466	0.407	0.324	0.378
SC-MOGA with data sampling	0.892	0.882	0.990	0.937	0.955	0.931

Table 6 *(Continue)*

Dataset	**Number of clusters**					
Seeds	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.777	0.754	0.754	0.681	0.631	0.719
SC-MOGA	0.830	0.808	0.743	0.700	0.640	0.744
SC-MOGA with data sampling	1	1	0.868	0.806	0.783	0.891
SPECTF	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.074	0.136	0.133	0.137	0.164	0.129
SC-MOGA	0.079	0.143	0.412	0.374	0.341	0.270
SC-MOGA with data sampling	1	1	1	1	1	1
SPF	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.398	0.407	0.385	0.348	0.366	0.381
SC-MOGA	0.244	0.184	0.083	0.009	0.009	0.106
SC-MOGA with data sampling	0.664	0.765	0.776	0.709	0.719	0.727
TAE	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.145	0.140	0.140	0.178	0.158	0.152
SC-MOGA	0.444	0.538	0.446	0.604	0.558	0.518
SC-MOGA with data sampling	0.655	0.638	0.555	0.696	0.649	0.639
Vertebral	**K=3**	**K=6**	**K=9**	**K=12**	**K=15**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.549	0.549	0.441	0.430	0.392	0.472
SC-MOGA	0.564	0.676	0.576	0.584	0.466	0.573
SC-MOGA with data sampling	0.779	0.724	0.618	0.764	0.789	0.735

Table 6 *(Continue)*

Dataset			**Number of clusters**			
Wilt	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.019	0.021	0.010	0.010	0.040	0.020
SC-MOGA	0.027	0.049	0.055	0.022	0.041	0.039
SC-MOGA with data sampling	0.908	0.870	1	0.921	0.850	0.910
Wine	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.927	0.809	0.772	0.691	0.631	0.766
SC-MOGA	0.973	0.823	0.753	0.688	0.641	0.776
SC-MOGA with data sampling	1	0.949	0.968	0.856	0.849	0.924

Table 7

Mirkin distance (MD) results on 23 datasets

Dataset			**Number of clusters**			
Iris	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	0.393	0.349	0.352	0.333	0.328	0.351
LK-Means	0.158	0.158	0.148	0.147	0.158	0.154
SCEC	0.034	0.107	0.148	0.165	0.198	0.130
SC-MOGA	0.034	0.086	0.136	0.178	0.187	0.124
SC-MOGA with data sampling	0	0	0.008	0.017	0.017	0.008
Heart	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.491	0.450	0.451	0.445	0.450	0.457
LK-Means	0.347	0.361	0.402	0.403	0.419	0.386
SCEC	0.237	0.317	0.406	0.389	0.421	0.354
SC-MOGA	0.118	0.283	0.375	0.397	0.428	0.320
SC-MOGA with data sampling	0.092	0.281	0.349	0.397	0.411	0.306

Table 7 *(Continue)*

Dataset	**Number of clusters**					
Glass	**K=6**	**K=7**	**K=8**	**K=9**	**K=10**	**Mean**
SRIDHCR	0.325	0.302	0.309	0.298	0.286	0.304
LK-Means	0.324	0.314	0.317	0.291	0.267	0.303
SCEC	0.230	0.245	0.213	0.219	0.208	0.223
SC-MOGA	0.211	0.195	0.207	0.206	0.215	0.207
SC-MOGA with data sampling	0.088	0.095	0.099	0.112	0.134	0.106
Diabetes	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	0.406	0.493	0.493	0.506	0.512	0.482
LK-Means	0.419	0.471	0.492	0.504	0.503	0.478
SCEC	0.351	0.467	0.470	0.507	0.509	0.461
SC-MOGA	0.315	0.386	0.461	0.494	0.502	0.432
SC-MOGA with data sampling	0.246	0.378	0.458	0.488	0.507	0.415
Vehicle Silhouettes	**K=4**	**K=8**	**K=12**	**K=16**	**K=20**	**Mean**
SRIDHCR	0.391	0.319	0.281	0.264	0.262	0.303
LK-Means	0.381	0.287	0.261	0.258	0.246	0.287
SCEC	0.289	0.229	0.225	0.228	0.235	0.241
SC-MOGA	0.207	0.201	0.215	0.205	0.216	0.209
SC-MOGA with data sampling	0.189	0.194	0.201	0.208	0.215	0.201
Image Segmentation	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	0.149	0.111	0.112	0.120	0.124	0.123
LK-Means	0.154	0.104	0.107	0.109	0.111	0.117
SCEC	0.070	0.077	0.081	0.086	0.091	0.081
SC-MOGA	0.123	0.126	0.134	0.137	0.143	0.133
SC-MOGA with data sampling	0.082	0.081	0.099	0.097	0.106	0.093
Ionosphere	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	0.429	0.420	0.450	0.445	0.474	0.444
LK-Means	0.398	0.403	0.440	0.411	0.444	0.419
SCEC	0.269	0.294	0.302	0.329	0.396	0.318
SC-MOGA	0.306	0.293	0.356	0.442	0.464	0.372
SC-MOGA with data sampling	0.185	0.257	0.332	0.372	0.387	0.307

Table 7 *(Continue)*

Dataset	Number of clusters					
Sonar	**K=2**	**K=4**	**K=6**	**K=8**	**K=10**	**Mean**
SRIDHCR	0.479	0.506	0.474	0.503	0.480	0.488
LK-Means	0.455	0.460	0.458	0.444	0.440	0.451
SCEC	0.351	0.361	0.389	0.399	0.413	0.383
SC-MOGA	0.212	0.222	0.292	0.347	0.378	0.290
SC-MOGA with data sampling	0	0.005	0.029	0.005	0.005	0.009
BS	**K=3**	**K=7**	**K=11**	**K=15**	**K=19**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.196	0.326	0.359	0.372	0.385	0.328
SC-MOGA	0.170	0.328	0.362	0.384	0.394	0.328
SC-MOGA with data sampling	0.169	0.316	0.359	0.372	0.383	0.320
BTSC	**K=2**	**K=7**	**K=12**	**K=17**	**K=22**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.343	0.343	0.586	0.581	0.596	0.490
SC-MOGA	0.501	0.523	0.561	0.598	0.614	0.559
SC-MOGA with data sampling	0.242	0.258	0.267	0.280	0.270	0.263
CMSC	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.501	0.718	0.735	0.772	0.778	0.701
SC-MOGA	0.388	0.632	0.740	0.794	0.805	0.672
SC-MOGA with data sampling	0.102	0.194	0.215	0.224	0.244	0.196
CMC	**K=3**	**K=9**	**K=15**	**K=21**	**K=27**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.439	0.375	0.361	0.360	0.358	0.379
SC-MOGA	0.361	0.353	0.354	0.365	0.363	0.359
SC-MOGA with data sampling	0.189	0.043	0.144	0.237	0.232	0.169

Table 7 *(Continue)*

Dataset	**Number of clusters**					
HS	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.350	0.476	0.509	0.544	0.565	0.489
SC-MOGA	0.499	0.424	0.480	0.507	0.523	0.487
SC-MOGA with data sampling	0.332	0.225	0.199	0.243	0.213	0.242
LD	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.455	0.459	0.468	0.478	0.471	0.466
SC-MOGA	0.335	0.266	0.369	0.410	0.431	0.362
SC-MOGA with data sampling	0.310	0.144	0.091	0.073	0.129	0.149
MP	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.376	0.311	0.404	0.430	0.429	0.390
SC-MOGA	0.243	0.325	0.400	0.427	0.448	0.369
SC-MOGA with data sampling	0.176	0.189	0.246	0.247	0.283	0.228
Musk	**K=2**	**K=6**	**K=10**	**K=14**	**K=18**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.409	0.458	0.459	0.450	0.459	0.447
SC-MOGA	0.361	0.350	0.395	0.432	0.456	0.399
SC-MOGA with data sampling	0.033	0.038	0.002	0.016	0.011	0.020
Seeds	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.078	0.096	0.096	0.146	0.197	0.123
SC-MOGA	0.068	0.105	0.165	0.180	0.226	0.149
SC-MOGA with data sampling	0	0	0.069	0.112	0.117	0.060

Table 7 *(Continue)*

Dataset	**Number of clusters**					
SPECTF	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.305	0.537	0.558	0.591	0.611	0.520
SC-MOGA	0.448	0.592	0.512	0.536	0.566	0.531
SC-MOGA with data sampling	0	0	0	0	0	0
SPF	**K=7**	**K=14**	**K=21**	**K=28**	**K=35**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.226	0.225	0.190	0.209	0.203	0.211
SC-MOGA	0.235	0.230	0.238	0.241	0.241	0.237
SC-MOGA with data sampling	0.188	0.093	0.088	0.118	0.117	0.121
TAE	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.386	0.352	0.353	0.324	0.326	0.348
SC-MOGA	0.257	0.211	0.287	0.206	0.231	0.238
SC-MOGA with data sampling	0.156	0.175	0.282	0.152	0.183	0.190
Vertebral	**K=3**	**K=6**	**K=9**	**K=12**	**K=15**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.165	0.165	0.283	0.283	0.308	0.241
SC-MOGA	0.261	0.198	0.260	0.269	0.309	0.259
SC-MOGA with data sampling	0.134	0.165	0.255	0.151	0.144	0.170
Wilt	**K=2**	**K=5**	**K=8**	**K=11**	**K=14**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.498	0.682	0.803	0.828	0.836	0.729
SC-MOGA	0.436	0.757	0.835	0.871	0.830	0.746
SC-MOGA with data sampling	0.003	0.005	0	0.004	0.006	0.004

Table 7 *(Continue)*

Dataset	**Number of clusters**					
Wine	**K=3**	**K=5**	**K=7**	**K=9**	**K=11**	**Mean**
SRIDHCR	N/A	N/A	N/A	N/A	N/A	N/A
LK-Means	N/A	N/A	N/A	N/A	N/A	N/A
SCEC	0.024	0.111	0.137	0.190	0.231	0.139
SC-MOGA	0.008	0.106	0.148	0.193	0.171	0.125
SC-MOGA with data sampling	0	0.025	0.013	0.085	0.071	0.039

Table 8

Confidence levels of paired t-test between SC-MOGA and the other three algorithms based on the eight datasets and different numbers of clusters

	SRIDHCR	**LK-Means**	**SC-MOGA with data sampling**
Based on AMI, SC-MOGA performs better than	100%	100%	0%
Based on ARI, SC-MOGA performs better than	100%	100%	0%
Based on AVI, SC-MOGA performs better than	100%	100%	0%
Based on MD, SC-MOGA performs better than	100%	100%	0%

Table 9

Confidence levels of paired t-test between SC-MOGA with data sampling and the other three algorithms based on the eight datasets and different numbers of clusters

	SRIDHCR	**LK-Means**	**SC-MOGA**
Based on AMI, SC-MOGA with data sampling performs better than	100%	100%	100%
Based on ARI, SC-MOGA with data sampling performs better than	100%	100%	100%
Based on AVI, SC-MOGA with data sampling performs better than	100%	100%	100%
Based on MD, SC-MOGA with data sampling performs better than	100%	100%	100%

The Second Experiment

The second experiment is intended to compare the performances of SC-MOGA and SC-MOGA with data sampling against SCEC, which is also based on a genetic algorithm. The testing datasets comprise the eight datasets used in the first experiment and another fifteen datasets. These fifteen datasets are also from the UCI Machine Learning Repository (Lichman, 2013). They are Balance Scale ('BS'), Blood Transfusion Service Center ('BTSC'), Climate Model Simulation Crashes ('CMSC'), Contraceptive Method Choice ('CMC'), Haberman's Survival ('HS'), Liver Disorders ('LD'), MONK's Problems ('MP'), Musk (Version 1) ('Musk'), Seeds, SPECTF Heart ('SPECTF'), Steel Plates Faults ('SPF'), Teaching Assistant Evaluation ('TAE'), Vertebral, Wilt and Wine. All datasets were preprocessed by max-min normalization. The details of the fifteen datasets are shown in Table 10. The setups for the second experiment were the same as those for the first experiment.

Table 11 shows all the sampling parameter values used in the experiments for SC-MOGA with data sampling. We implemented SCEC and ran it based on the experiment setups in Eick, Zeidat and Zhao (2004): size of population is 400, Crossover rate increases from 0 to 0.95, Mutation rate decreases from 0.95 to 0, Number of generations is 1,500, Copy rate is 0.05. The value of the parameter α in equation 1 was chosen from the 11 values between 0 and 2.0 with a step of 0.2, which yielded the best performance. The four cluster validity indexes were used to compare the performance. The four validity index results for the 23 datasets are shown in Tables 4 to 7. One-sided paired t-tests were carried out to compare the performances of SC-MOGA, SC-MOGA with data sampling and SCEC. To carry out the t-test between two algorithms for each index, we computed the differences between the indexes achieved by the two algorithms for the 23 datasets and for all Ks (number of clusters). The results of the tests are shown in Tables 12 and 13. The results in Table 12 show that SC-MOGA achieved better performances than SCEC with a confidence level of more than 95% for all four indexes, but did not perform better than SC-MOGA with data sampling. The results in Table 13 show that SC-MOGA with data sampling achieved a better performance than SCEC and SC-MOGA with confidence levels of more than 95% for all four indexes.

It can be concluded from the t-test results for the two experiments that the proposed SC-MOGA and SC-MOGA with data sampling methods achieved better performances than the other three algorithms in existence. It can also be seen from the results that SC-MOGA with data sampling achieved better performances than SC-MOGA. For the sake of brevity, the plots of AMI against running time in seconds on the eight datasets for SCEC, SC-MOGA and SC-MOGA with data sampling are shown in Figures 2 to 9. The plots for the other datasets exhibit similar results. It can be seen from the figures that although SCEC can converge quickly, it experienced premature convergences to local optima, while

SC-MOGA and SC-MOGA with data sampling took more time but were able to converge to better solutions. It can also be seen that SC-MOGA with data sampling took less time than SC-MOGA to converge, and that it converged to better solutions. This shows that the proposed sampling method is very effective for sampling good representatives of the given dataset, reducing the size of search space and allowing the genetic algorithm to converge to better solutions.

For a very large dataset, the search space for the proposed genetic algorithm can be huge, therefore, SC-MOGA or SC-MOGA with data sampling may take quite a large number generations to converge to solutions. Some future work could be done to further improve the performance of the proposed algorithm. For example, to help the algorithm converge more quickly it is possible to incorporate specialized genetic operators which perform some local search for offspring with fitness values better than their parent chromosomes. This would accelerate the search to converge more quickly toward the potential optimal solution. Some other evolutionary algorithms such as particle swarm

Table 10

Details of the fifteen UCI Machine Learning Repository Datasets

Dataset Name	# objects	# variables	# classes
BS	625	4	3
BTSC	748	4	2
CMSC	540	18	2
CMC	1473	9	3
HS	306	3	2
LD	345	6	2
MP	432	6	2
Musk	476	166	2
Seeds	210	7	3
SPECTF	267	44	2
SPF	1941	27	7
TAE	151	5	3
Vertebral	310	6	3
Wilt	4339	5	2
Wine	178	13	3

Table 11

Sampling parameter values used by SC-MOGA with data sampling on the additional 15 datasets

Dataset	Number of Data Objects (N)	χ^2 chi-square	Population Fraction (F)	Precision Degree (d)	Expected Sample Size (S)	Number of Strata (m)	Actual Sample size $\sum_{k=1}^{m} n_k$.
BS	625	3.841	0.5	0.05	239	322	431
BTSC	748	3.841	0.5	0.05	255	368	495
CMSC	540	3.841	0.5	0.05	225	183	289
CMC	1473	3.841	0.5	0.05	305	43	306
HS	306	3.841	0.5	0.05	171	144	216
LD	345	3.841	0.5	0.05	183	109	212
MP	432	3.841	0.5	0.05	204	150	211
Musk	476	3.841	0.5	0.05	213	10	213
Seeds	210	3.841	0.5	0.05	136	14	136
SPECTF	267	3.841	0.5	0.05	158	16	158
SPF	1941	3.841	0.5	0.05	321	38	321
TAE	151	3.841	0.5	0.05	109	41	109
Vertebral	310	3.841	0.5	0.05	172	14	172
Wilt	4339	3.841	0.5	0.05	353	12	353
Wine	178	3.841	0.5	0.05	122	12	122

intelligence have proven to be very effective for solving large dimensional optimization problems, so it is possible to use these algorithms for searching clustering solutions instead of the genetic algorithm.

Table 12

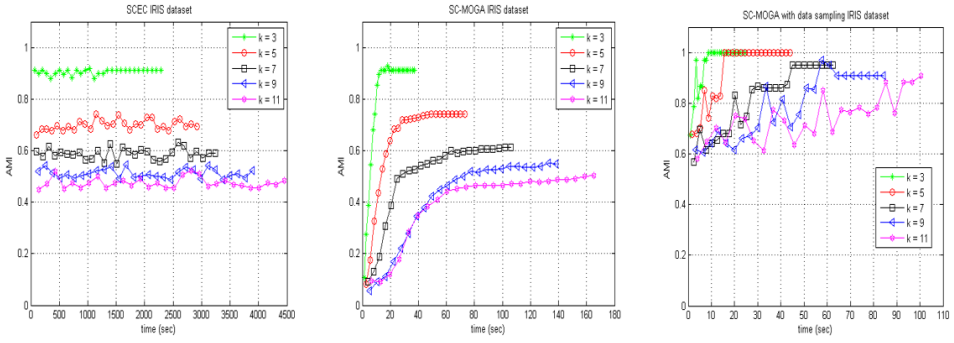
Confidence levels of paired t-test based on the four indexes between SC-MOGA and the other two algorithms based on the 23 datasets and different numbers of clusters

	SCEC	**SC-MOGA with data sampling**
Based on AMI, SC-MOGA performs better than	100%	0%
Based on ARI, SC-MOGA performs better than	99%	0%
Based on AVI, SC-MOGA performs better than	100%	0%
Based on MD, SC-MOGA performs better than	98%	0%

Table 13

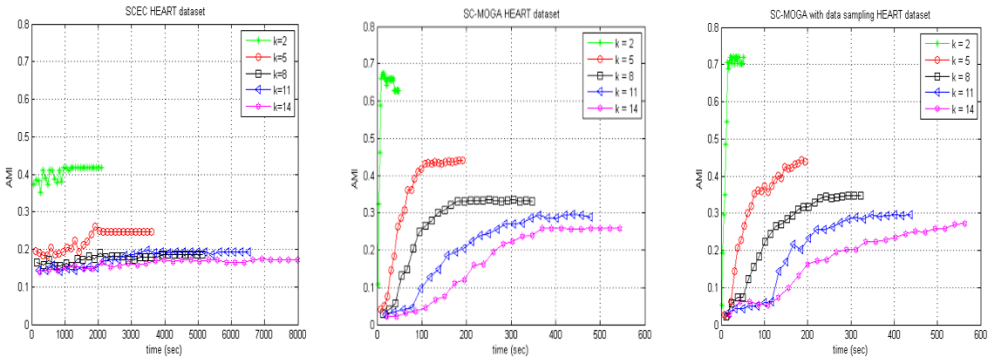
Confidence levels of paired t-test based on the four indexes between the SC-MOGA with data sampling and the other two algorithms, based on the 23 datasets and different numbers of clusters.

	SCEC	**SC-MOGA**
Based on AMI, SC-MOGA with data sampling performs better than	100%	100%
Based on ARI, SC-MOGA with data sampling performs better than	100%	100%
Based on AVI, SC-MOGA with data sampling performs better than	100%	100%
Based on MD, SC-MOGA with data sampling performs better than	100%	100%



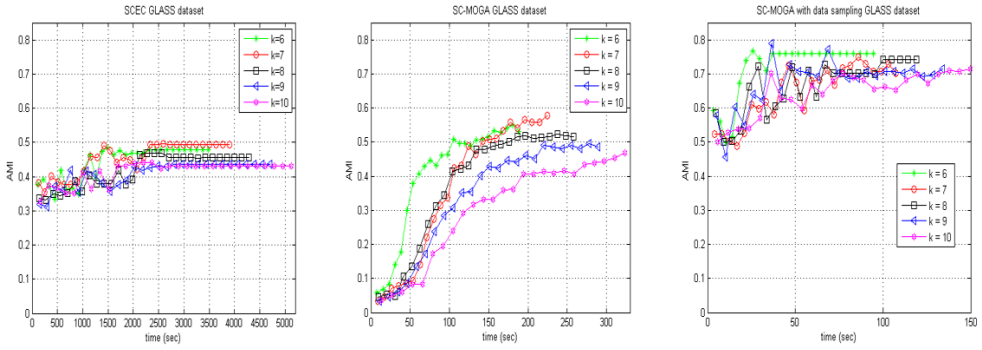
(a) (b) (c)

Figure 2. (a), (b), (c) Plots of AMI against running time in seconds on Iris for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively



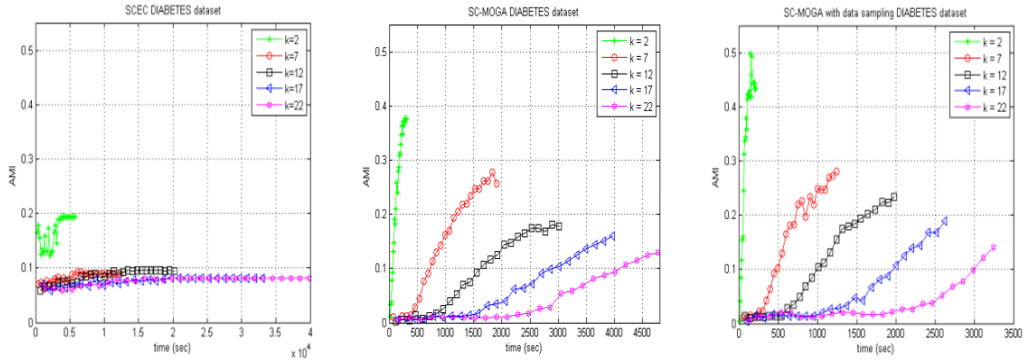
(a) (b) (c)

Figure 3. (a), (b), (c) Plots of AMI against running time in seconds on Heart for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively



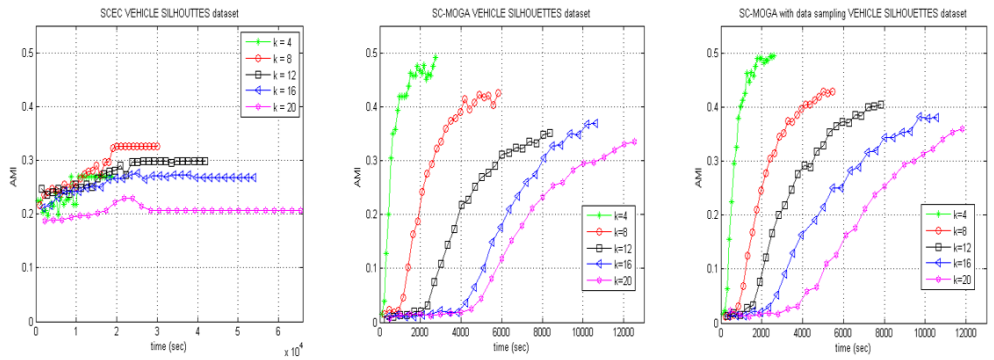
(a) (b) (c)

Figure 4. (a), (b), (c) Plots of AMI against running time in seconds on Glass for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively



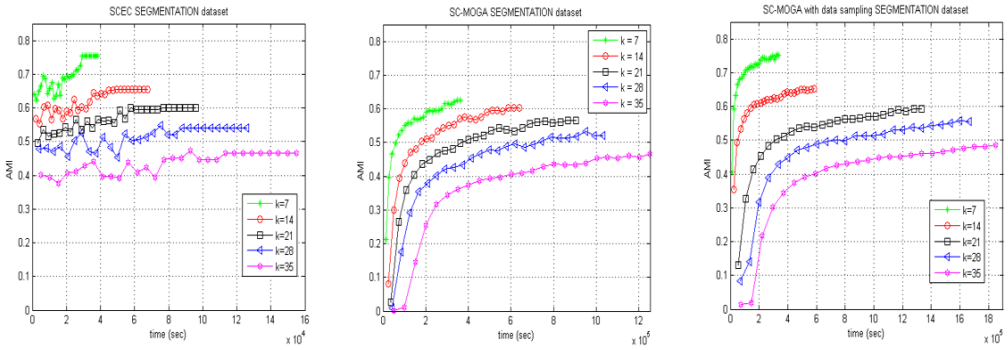
(a) (b) (c)

Figure 5. (a), (b), (c) Plots of AMI against running time in seconds on Diabetes for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively



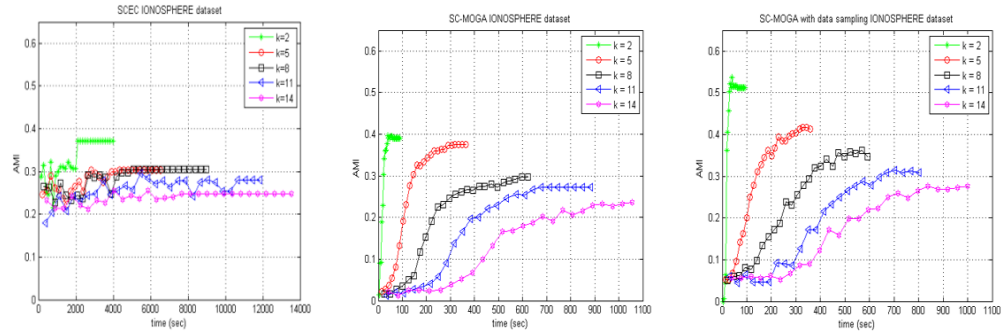
(a) (b) (c)

Figure 6. (a), (b), (c) Plots of AMI against running time in seconds on Vehicle Silhouettes for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively



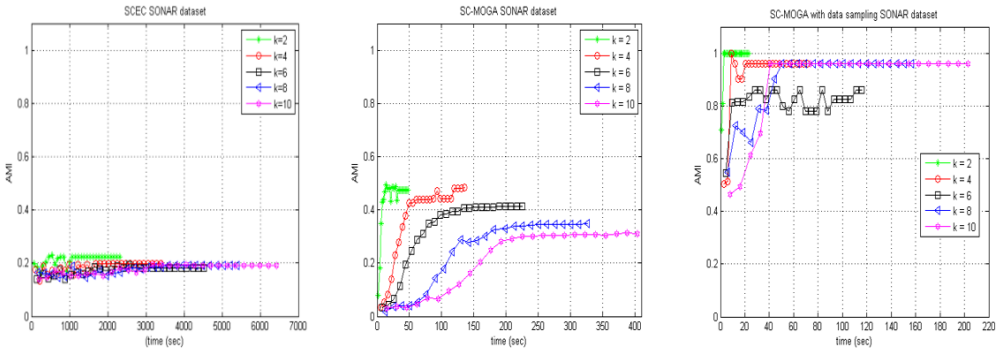
(a) (b) (c)

Figure 7. (a), (b), (c) Plots of AMI against running time in seconds on Segmentation for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively



(a) (b) (c)

Figure 8. (a), (b), (c) Plots of AMI against running time in seconds on Ionosphere for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively



(a) (b) (c)

Figure 9. (a), (b), (c) Plots of AMI against running time on Sonar for SCEC, SC-MOGA and SC-MOGA with data sampling, respectively

Convergence of the Three Objective Functions

Figures 10 and 11 show the convergence of the normalized values (using a max-min normalization scheme) of the three objective functions, impurity level, SSE and inter cluster distance, on the Iris dataset for SC-MOGA and SC-MOGA with data sampling. The convergence of the normalized values of the three objective functions on the other testing datasets exhibit quite the same patterns so they are not included in this paper. The convergence plots of AMI for SC-MOGA and SC-MOGA with data sampling are also shown in figure 10(a) and figure 11(a) for comparison with those of the three objective functions. It can be seen from the figures 10(a) and 11(a) - 10(f) and 11(f) that when the number of generations of the genetic algorithms increases, the SSE and the impurity level decreases while the inter cluster distance (as well as the AMI) increases until they all converge. The results show that the proposed algorithms can simultaneously optimize the three objective functions leading to a good clustering solution as measured by the AMI value. It can also be seen that as the number of clusters, K, increases, it takes more number of generations for the genetic algorithms to converge to an optimal solution. This is due the fact that the search space for the genetic algorithms becomes more complex as the value of K increases.

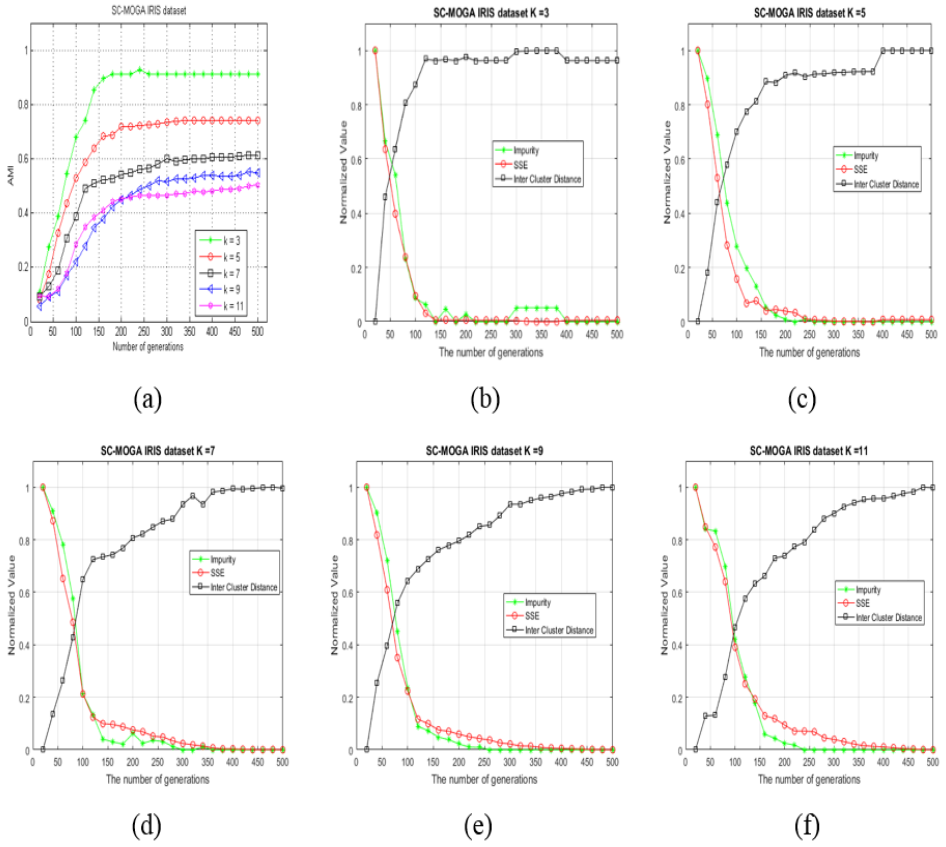


Figure 10. (a) Plots of AMI on Iris within 500 generations for SC-MOGA when K = 3, 5, 7, 9 and 11. (b) – (f) Plots of normalized values of impurity, SSE and inter cluster distance within 500 generations for SC-MOGA when K= 3, 5, 7, 9 and 11, respectively.

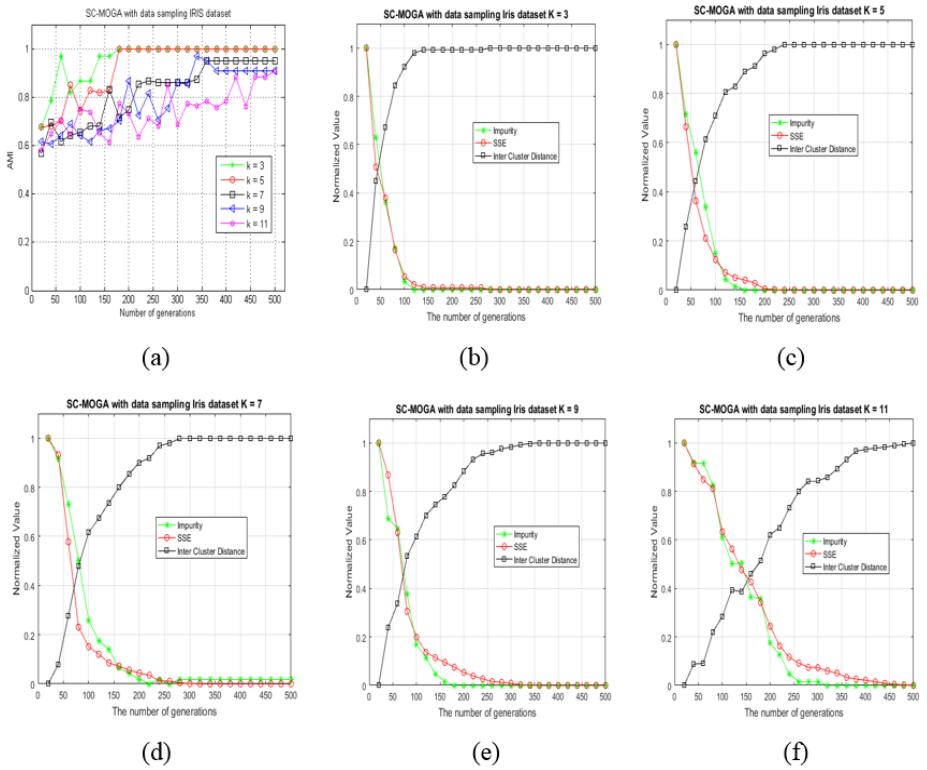


Figure 11. (a) Plots of AMI on Iris within 500 generations for SC-MOGA with data sampling when K = 3, 5, 7, 9 and 11. (b) – (f) Plots of normalized values of impurity, SSE and inter cluster distance within 500 generations for SC-MOGA with data sampling when K= 3, 5, 7, 9 and 11, respectively.

CONCLUSION

A novel supervised clustering algorithm based on a multi-objective genetic algorithm, namely SC-MOGA, is proposed in this paper. The SC-MOGA incorporates three objectives for supervised clustering. The first objective is to minimize the sum squared errors (compactness) of the clusters, the second objective is to minimize the level of impurity of the data in the clusters and the third objective is to maximize the inter cluster distance (separateness). The SC-MOGA applies the crowding genetic algorithm to search for the clustering solutions in a multimodal solution space. For large datasets, a data sampling method using the Bisecting K-Means approach is also proposed to sample the representatives of the dataset for clustering. The experimental results show that the SC-MOGA and SC-MOGA with data sampling are very effective for supervised clustering. They outperform some existing algorithms, i.e. LK-Means, SRIDHCR and SCEC. The experiment results also show that the proposed data sampling method not only helps reduce the sample size for SC-MOGA but also helps SC-MOGA to converge to better clustering solutions.

REFERENCES

Cadima, E. L., Caramelo, A. M., Afonso-Dias, M., Conte de Barros, P., Tandstad, M. O., & de Leiva Moreno, J. I. (2005). *Sampling methods applied to fisheries science: a manual* (FAO Fisheries Technical Paper No. 434). Retrieved September 3, 2016, from http://www.fao.org/docrep/009/a0198e/A0198E00.htm

De Jong, K. A. (1975). *An analysis of the behaviour of a class of genetic adaptive systems* (PhD thesis). Department of Computer and Communication Sciences, University of Michigan, MI.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY: John Wiley & Sons, Inc.

Eick, C. F., Rouhana, A., Bagherjeiran, A., & Vilalta, R. (2006). Using clustering to learn distance functions for supervised similarity assessment. *Engineering Applications of Artificial Intelligence, 19*(4), 395–401.

Eick, C. F., Vaezian, B., Jiang, D., & Wang, J. (2006). Discovery of interesting regions in spatial data sets using supervised clustering. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 127–138). Berlin, Germany.

Eick, C. F., Zeidat, N., & Vilalta, R. (2004). Using representative-based clustering for nearest neighbor dataset editing. In *Proceedings of the 4th IEEE International Conference on Data Mining* (pp. 375–378). DC, USA.

Eick, C. F., Zeidat, N., & Zhao, Z. (2004). Supervised clustering – algorithms and benefits. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence* (pp.774-776). DC, USA.

Finley, T., & Joachims, T. (2005). Supervised clustering with support vector machines. In *Proceedings of the 22nd International Conference on Machine Learning* (pp.17–224). NY, USA.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 416-423). CA, USA.

Grbovic, M., Djuric, N., Guo, S., & Vucetic, S. (2013). Supervised clustering of label ranking data using label preference information. *Journal Machine Learning*, *93*(2–3), 191–225.

Haider, P., Brefeld, U., & Scheffer, T. (2007). Supervised clustering of streaming data for email batch detection. In *Proceedings of the 24th International Conference on Machine Learning* (pp. 345–352). OR, USA.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal, 29*(2), 147–160. doi: 10.1002/j.1538-7305.1950.tb00463.x

Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., & de Carvalho, A. C. P. L. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 39*(2), 133–155. doi: 10.1109/TSMCC.2008.2007252

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classifications, 2*(1), 193–218. doi: 10.1007/BF01908075

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Hoboken: John Wiley & Sons. doi: 10.1002/9780470316801

Krejcie, R. V., & Morgan, D. W. (1970). Determining sample size for research activities. *Educational and Psychological Measurement, 30*(3), 607–610.

Lichman, M. (2013). *UCI Machine Learning Repository*. Retrieved September 3, 2016, from http://archive.ics.uci.edu/ml

Maji, P. (2010). Mutual information-based supervised attribute clustering for microarray sample classification. *IEEE Transactions on Knowledge and Data Engineering, 24*(1), 127–140. doi:10.1109/TKDE.2010.210

Mirkin, B. G., & Chernyj, L. B. (1970). Measurement of the distance between distinct partitions of a finite set of objects. *Automation and Remote Control, 5*, 786–792.

Peralta, B., Espinace, P., & Soto, A. (2013). Enhancing K-Means using class labels. *Journal Intelligent Data Analysis, 17*(6), 1023–1039.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms* (pp. 2-9). CA, USA.

Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 1073-1080). Quebec, Canada.