Pattern Recognition 43 (2010) 738–751

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A symmetry based multiobjective clustering technique for automatic evolution of clusters

Sriparna Saha*, Sanghamitra Bandyopadhyay

Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India

ARTICLE INFO

Article history:
Received 7 November 2008
Received in revised form 16 June 2009
Accepted 6 July 2009

Keywords:
Clustering
Multiobjective optimization (MOO)
Symmetry
Point symmetry based distance
Cluster validity index
Simulated annealing (SA)

ABSTRACT

In this paper the problem of automatic clustering a data set is posed as solving a multiobjective optimization (MOO) problem, optimizing a set of cluster validity indices simultaneously. The proposed multiobjective clustering technique utilizes a recently developed simulated annealing based multiobjective optimization method as the underlying optimization strategy. Here variable number of cluster centers is encoded in the string. The number of clusters present in different strings varies over a range. The points are assigned to different clusters based on the newly developed point symmetry based distance rather than the existing Euclidean distance. Two cluster validity indices, one based on the Euclidean distance, XB-index, and another recently developed point symmetry distance based cluster validity index, *Sym*-index, are optimized simultaneously in order to determine the appropriate number of clusters present in a data set. Thus the proposed clustering technique is able to detect both the proper number of clusters and the appropriate partitioning from data sets either having hyperspherical clusters or having point symmetric clusters. A new semi-supervised method is also proposed in the present paper to select a single solution from the final Pareto optimal front of the proposed multiobjective clustering technique. The efficacy of the proposed algorithm is shown for seven artificial data sets and six real-life data sets of varying complexities. Results are also compared with those obtained by another multiobjective clustering technique, MOCK, two single objective genetic algorithm based automatic clustering techniques, VGAPS clustering and GCUK clustering.

1. Introduction

Clustering is the task of finding natural partitioning within a data set such that data items within the same group are more similar than those within different groups. Often for many data sets, no unambiguous partitioning of the data exists. Moreover, most existing clustering algorithms are based only on one internal evaluation function. This is an objective function which measures some intrinsic property of a partitioning. These may be the spatial separation between the clusters, the compactness of the clusters, and a measure of cluster symmetry. Thus the internal evaluation function is assumed to reflect the quality of the partitioning reliably. But this assumption may be violated for many data sets. In view of the above difficulties in defining the notion of "appropriateness" of clustering solutions, application of multiobjective optimization (MOO) techniques appears to be an alternative and promising direction [1]. Rather than optimizing a single criterion, it may be better to optimize *multiple* proxy measures of cluster quality—namely cluster validity indices. In this article the problem of automatically partitioning a data set is posed as one of the multiobjective optimizations (MOO) [2]. In contrast to single objective optimization, which yields a single best solution, in MOO the final solution set contains a number of Pareto optimal solutions, none of which can be further improved on any one objective without degrading it in another.

For identifying clusters from a data set, at first some measure of similarity or proximity has to be defined. Then this similarity measure is used to assign points to different clusters. In general, many clusters are symmetrical with respect to some axis or point. Based on this observation, a new point symmetry (PS) based distance, d_{ps} (PS-distance), is developed in [3]. For reducing the complexity of computing the PS-distance, the use of Kd-tree is incorporated in [3].

Determining the appropriate number of clusters from a given data set is an important consideration in clustering. For this purpose, and also to validate the obtained partitioning, several cluster validity indices have been proposed in the literature. The measure of validity of the clusters should be such that it will be able to impose an

* Corresponding author.
E-mail addresses: sriparna_r@isical.ac.in, sriparna.saha@gmail.com (S. Saha), sanghami@isical.ac.in (S. Bandyopadhyay).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.07.004

ordering of the clusters in terms of its goodness. Since the global optimum of the validity functions would correspond to the most "valid" solutions with respect to the functions, stochastic clustering algorithms based on genetic algorithms (GAs) have been reported to optimize the validity functions to determine the number of clusters and the partitioning of a data set simultaneously [4–6]. Other than evaluating the static clusters generated by a specific clustering algorithm, the validity functions in these approaches are used as clustering objective functions for computing the fitness, which guides the evolution to search for the "valid" solution. However, simple GA (SGA) [7] or its variants are used as the genetic clustering techniques in [4–6]. In [8], a function called weighted sum validity function (WSVF), which is a weighted sum of the several normalized validity functions, is used for optimization along with a hybrid niching genetic algorithm (HNGA) to automatically evolve the proper number of clusters from a given data set. Within this HNGA, a niching method is developed to prevent premature convergence by preserving both the diversity of the population with respect to the number of clusters encoded in the individuals and the diversity of the subpopulation with the same number of clusters during the search. In [9], a multiobjective genetic approach is used for clustering where several validity functions are simultaneously optimized.

But in the above mentioned genetic clustering techniques for automatic evolution of clusters, assignment of points to different clusters is done in the lines of *K*-means clustering algorithm. Consequently, all these approaches are only able to find compact hyperspherical, equisized and convex clusters like those detected by the *K*-means algorithm [10]. If clusters of different geometric shapes are present in the same data set, the above methods will not be able to find all of them perfectly. In [11] an attempt had been taken in this direction. A point symmetry based genetic clustering technique is proposed in [11], named VGAPS clustering, for automatic determination of the number of clusters and the appropriate partitioning from data sets having point symmetric clusters. VGAPS uses the newly developed point symmetry based distance [3] for assignment of points. But it optimizes a single cluster validity measure, point symmetry based cluster validity index, *Sym*-index, as the fitness function to reflect the goodness of an encoded partitioning. However, a single cluster validity measure like *Sym*-index is seldom equally applicable for different kinds of data sets with different characteristics. Hence it is necessary to simultaneously optimize several validity measures that can capture the different data characteristics. In order to achieve this, in this article the problem of clustering a data set is posed as one of the multiobjective optimizations (MOO) [2], where search is performed over a number of, often conflicting, objective functions. A newly developed simulated annealing based multiobjective optimization technique, AMOSA [12], is used in this paper to determine the appropriate cluster centers and the corresponding partitioning from a data set.

A multiobjective clustering technique, MOCK, is developed in [1] which can automatically detect the appropriate partitioning from a data set having either the hyperspherical shaped clusters or the well-separated clusters. But it fails to detect overlapping clusters having different shapes other than hyperspheres. Another important drawback of MOCK is in its encoding. Here locus based adjacency representation [13] is used. Thus the length of each chromosome is equal to the number of points present in the data set. Hence it will increase largely with the increase in the number of points.

In this paper a new multiobjective clustering technique, VAMOSA, is proposed where the center based encoding is used. Here strings encode the centers of a variable number of clusters where their values vary over a range. Points are assigned to different clusters based on the newly developed point symmetry based distance [3] rather than the Euclidean distance. This enables the proposed clustering technique to automatically evolve the appropriate partitioning from data sets having clusters of any shape, size or convexity as long as they do possess some point symmetry property. Two cluster validity indices are optimized simultaneously, a well-known Euclidean distance based XB-index [14], and another recently developed point symmetry distance based *Sym*-index [15,11]. Note that any other and any number of objective functions could be used instead of the above mentioned two. AMOSA [12], a recently developed simulated annealing based multiobjective optimization technique, is used here as the underlying optimization strategy. The effectiveness of the proposed symmetry based automatic multiobjective clustering technique (VAMOSA) is shown for seven artificial and six real-life data sets of varying complexities. Comparisons are also made with another multiobjective (MO) clustering technique, MOCK [1], two state-of-the-art single objective automatic clustering techniques, VGAPS clustering [11] optimizing a newly developed symmetry based cluster validity index, *Sym*-index, and GCUK clustering [6] optimizing XB-index [16]. In the present paper some statistical analyses in the light of [17,18] are also done.

This paper is organized as follows. Section 2 discusses the existing multiobjective simulated annealing based optimization technique, AMOSA. Section 3 describes in detail the existing point symmetry (PS) distance and the use of Kd-tree based approximate nearest neighbor search for reducing time complexity of computing PS based distance. A description of the newly proposed automatic multiobjective clustering technique with point symmetry based distance (VAMOSA) is given in Section 4, and Section 5 discusses the data sets used for experiment. Section 6 provides the experimental results. Section 7 concludes this paper.

2. The SA based MOO algorithm: AMOSA

Archived multiobjective simulated annealing (AMOSA) [12] is a generalized version of the simulated annealing (SA) algorithm based on multiobjective optimization (MO). MO is applied when dealing with the real-world problems where there are several objectives that should be optimized simultaneously. In general, a MO algorithm usually admits a set of solutions that are not dominated by any solution it encountered, i.e., non-dominated solutions [2]. During recent years, many multiobjective evolution algorithms, such as multiobjective EA (MOEA), have been suggested to solve the MO problems [19].

Simulated annealing (SA) is a search technique for solving difficult optimization problems, which is based on the principles of statistical mechanics [20]. Recently, SA has become very popular because not only can SA replace the exhaustive search to save time and resource, but also converge to the global optimum if annealed sufficiently slowly [21].

Although the single objective version of SA is quite popular, its utility in the multiobjective case was limited because of its search-from-a-point nature. Recently Bandyopadhyay et al. developed an efficient multiobjective version of SA called AMOSA [12] that overcomes this limitation. AMOSA is utilized in this work for partitioning a data set.

The AMOSA algorithm incorporates the concept of an archive where the non-dominated solutions seen so far are stored. Two limits are kept on the size of the archive: a hard or strict limit denoted by *HL* and a soft limit denoted by *SL*. The algorithm begins with the initialization of a number ($\gamma \times SL$, $\gamma > 1$) of solutions each of which represents a state in the search space. The multiple objective functions are computed. Each solution is refined by using simple hill-climbing and domination relation for a number of iterations. Thereafter the non-dominated solutions are stored in the archive until the size of the archive increases to *SL*. If the size of the archive exceeds *SL*, a single-linkage clustering scheme is used to reduce the

size to *HL*. Then, one of the points is randomly selected from the archive. This is taken as the current-pt, or the initial solution, at temperature $T=Tmax$. The current-pt is perturbed to generate a new solution named new-pt, and its objective functions are computed. The domination status of the new-pt is checked with respect to the current-pt and the solutions in the archive. A new quantity called amount of domination, $\Delta dom(a,b)$ between two solutions a and b, is defined as follows: $\Delta dom(a,b)=\prod_{i=1,f_i(a)\neq f_i(b)}^{M}(f_i(a)-f_i(b))/R_i$, where $f_i(a)$ and $f_i(b)$ are the i th objective values of the two solutions and R_i is the corresponding range of the objective function. Based on domination status different cases may arise viz., accept the (i) new-pt, (ii) current-pt, or (iii) a solution from the archive. Again, in case of overflow of the archive, clustering is used to reduce its size to *HL*. The process is repeated *iter* times for each temperature that is annealed with a cooling rate of $\alpha(<1)$ till the minimum temperature *Tmin* is attained. The process thereafter stops, and the archive contains the final non-dominated solutions.

It has been demonstrated in Ref. [12] that the performance of AMOSA is better than that of NSGA-II [22] and some other well-known MO algorithms.

3. The point symmetry distance

In this section at first the definition of the PS-distance developed in [3] is provided. Then, the use of Kd-tree for point symmetry distance computation is described.

3.1. Definition

A new definition of point symmetry based distance (PS-distance), $d_{ps}(\bar{x},\bar{c})$, associated with point $\bar{x}$ with respect to a center $\bar{c}$ is developed in Ref. [3]. It is also shown in [3] that $d_{ps}(\bar{x},\bar{c})$ is able to overcome some serious limitations of an earlier PS-distance [23]. The point symmetry based distance, $d_{ps}(\bar{x},\bar{c})$, is calculated in the following way. Let a point be $\bar{x}$. The symmetrical (reflected) point of $\bar{x}$ with respect to a particular center $\bar{c}$ is $2\times\bar{c}-\bar{x}$. Let us denote this by $\bar{x}^*$. Let *knear* unique nearest neighbors of $\bar{x}^*$ be at Euclidean distances of d_is, $i=1,2,\ldots,knear$. Then

$$d_{ps}(\bar{x},\bar{c})=d_{sym}(\bar{x},\bar{c})\times d_e(\bar{x},\bar{c}) \quad (1)$$

$$=\frac{\sum_{i=1}^{knear}d_i}{knear}\times d_e(\bar{x},\bar{c}), \quad (2)$$

where $d_e(\bar{x},\bar{c})$ is the Euclidean distance between the point $\bar{x}$ and $\bar{c}$, and $d_{sym}(\bar{x},\bar{c})$ is a symmetry measure of $\bar{x}$ with respect to $\bar{c}$.

The concept of point symmetry based distance is further illustrated by using Fig. 1. Here a particular point is $\bar{x}$. The cluster center is denoted by $\bar{c}$. Then the reflected point of $\bar{x}$ with respect to $\bar{c}$ is $\bar{x}'$, i.e., $\bar{x}'=2\times\bar{c}-\bar{x}$. The two nearest neighbors of $\bar{x}'$ are at an Euclidean distances of d_1 and d_2, respectively. Then the point symmetry based distance between $\bar{x}$ and $\bar{c}$ is calculated as $d_{ps}(\bar{x},\bar{c})=(d_1+d_2)/2\times d_e(\bar{x},\bar{c})$, where $d_e(\bar{x},\bar{c})$ is the Euclidean distance between the point $\bar{x}$ and the cluster center $\bar{c}$.

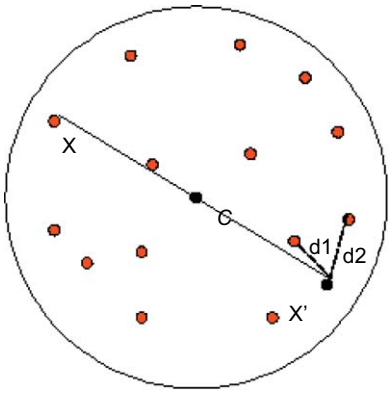

Fig. 1. Example of point symmetry distance.

Note that *knear* in Eq. (2) cannot be chosen equal to 1, since if $\bar{x}^*$ exists in the data set then $d_{ps}(\bar{x},\bar{c})=0$ and hence there will be no impact of the Euclidean distance. On the other hand, large values of *knear* may not be suitable because it may underestimate the amount of symmetry of a point with respect to a particular cluster center. Here *knear* is chosen equal to 2. It may be noted that the proper value of *knear* largely depends on the distribution of the data set. A fixed value of *knear* may have many drawbacks. For instance, for very large clusters (with too many points), two neighbors may not be enough as it is very likely that a few neighbors would have a distance close to zero. On the other hand, clusters with too few points are more likely to be scattered, and the distance of the two neighbors may be too large. Thus a proper choice of *knear* is an important issue that needs to be addressed in the future.

Note that $d_{ps}(\bar{x},\bar{c})$, which is a non-metric, is a way of measuring the amount of point symmetry between a point and a cluster center, rather than the distance like any Minkowski distance. The complexity of computing $d_{ps}(\bar{x},\bar{c})$ is of the order n, where n is the total number of data points. For all the n points and K clusters, the complexity becomes order n^2K. Thus, to reduce this, we have used Kd-tree based nearest neighbor search [24].

3.2. Kd-tree based nearest neighbor computation

A K-dimensional tree, or Kd-tree is a space-partitioning data structure for organizing points in a K-dimensional space. A Kd-tree uses only those splitting planes which are perpendicular to one of coordinate axes. In the nearest neighbor problem a set of data points in d-dimensional space is given. These points are preprocessed into a data structure, so that given any query point q, the nearest or generally k nearest points of p to q can be reported efficiently. ANN (approximate nearest neighbor) is a library written in C++ [25], which supports data structures and algorithms for both exact and approximate nearest neighbor searching in arbitrarily high dimensions. In this article ANN is used to find d_is in Eq. (2) efficiently. The ANN library implements a number of different data structures, based on Kd-trees and box-decomposition trees, and employs a couple of different search strategies. The Kd-tree data structure has been used in this article. ANN allows the user to specify a maximum approximation error bound, thus allowing the user to control the tradeoff between accuracy and running time.

For computing $d_{ps}(\bar{x},\bar{c})$ in Eq. (2), d_is need to be computed. This is a computation intensive task that can be speeded up by using the Kd-tree based nearest neighbor search. The function performing the k-nearest neighbor search in ANN is given a query point q (here it is $\bar{x}^*=2\times\bar{c}-\bar{x}$), a non-negative integer k (here it is set equal to *knear*), an array of point indices, nn_{idx}, and an array of distances, *dists*. Both arrays are assumed to contain at least k elements. This procedure computes the k nearest neighbors of q in the point set, and stores the indices of the nearest neighbors in the array nn_{idx}. Optionally a real value $\varepsilon\geq 0$ may be supplied. If so, then i th nearest neighbor is $(1+\varepsilon)$ approximation to the true i th nearest neighbor. That is, the true distance to this point may exceed the true distance to the real i th nearest neighbor of q by a factor of $(1+\varepsilon)$. If ε is omitted then the nearest neighbors will be computed exactly. For the purpose of this article, the exact nearest neighbor is computed; so the ε is set equal to 0. After getting the *knear* nearest neighbors of $\bar{x}^*$, the symmetrical distance of $\bar{x}$ with respect to a center $\bar{c}$ is calculated using Eq. (2). The Kd-tree structure can be constructed in $O(n\log n)$ time and takes $O(n)$ space.

4. Proposed method for multiobjective clustering

In this paper a new multiobjective clustering technique (VAMOSA) is proposed which uses the newly developed simulated annealing based MOO technique, AMOSA [12], as the underlying optimization strategy. The basic steps of VAMOSA, which closely follow those of the AMOSA optimization algorithm, are shown in Fig. 2. The clustering_PS() procedure is described in Fig. 3.

4.1. String representation and archive initialization

In AMOSA based clustering, the strings are made up of real numbers which represent the coordinates of the centers of the partitions. AMOSA attempts to evolve an appropriate set of cluster centers that represent the associated partitioning of the data. If a particular string encodes the centers of K clusters in d-dimensional space then its length l is taken to be $d * K$. For example, in four-dimensional space, the string ⟨2.3 1.4 7.6 12.9 2.1 3.4 0.01 12.2 0.06 2.3 6.7 15.3 3.2 11.72 9.5 3.4⟩ encodes four cluster centers (2.3, 1.4, 7.6, 12.9),(2.1, 3.4, 0.01, 12.2), (0.06, 2.3, 6.7, 15.3) and (3.2, 11.72, 9.5, 3.4).

Each center is considered to be indivisible. Each string i in the archive initially encodes the centers of a number, K_i, of clusters, such that $K_i = (rand() \bmod (K^{max} - 1)) + 2$. Here, $rand()$ is a function returning an integer, and K^{max} is a soft estimate of the upper bound of the number of clusters. The number of clusters will therefore ranges from two to K^{max}. The K_i centers encoded in a string are randomly selected distinct points from the data set.

Begin
1. $T = Tmax$; $Tmax$=maximum value of temperature.
2. initialize archive
3. $current\_sol$=randomly chosen solution from the archive
while $T < Tmin$; $Tmin$=minimum value of temperature.
4. For $i = 1$ to $iter$
call clustering_PS() procedure for $current\_sol$
compute objective functions for $current\_sol$
new_sol =mutate($current\_sol$)
call clustering_PS() procedure for new_sol
compute objective functions for new_sol
use the steps of AMOSA to decide who will be the next $current\_sol$,
whether to include new_sol in the archive etc.
$T = \alpha \times T$; α is the cooling rate.
end for
end While
5. select the best solution from the archive
6. output best solution and stop
End

Fig. 2. Basic steps of VAMOSA clustering technique.

4.2. Assignment of points

Let a particular string contain K number of clusters. The assignment of points to different clusters is done as in [3]. Each point $\overline{x}_j, j = 1, 2, \ldots n$, is assigned to a particular cluster in the following way. Find the cluster center nearest to $\overline{x_j}$ in the symmetrical sense. That is, we find the cluster center k that is nearest to the input pattern $\overline{x}_j$ using the minimum-value criterion: $k = Argmin_{i=1,\ldots,K} d_{ps}(\overline{x}_j, \overline{z}_i)$, where $\overline{z}_i$ denotes the center of the i th cluster and $d_{ps}(\overline{x}_j, \overline{z}_i)$ is the point symmetry based distance [3] between a particular point $\overline{x}_j$ and the cluster center $\overline{z}_i$. If the corresponding $d_{ps}(\overline{x}_j, \overline{z}_k)/d_e(\overline{x}_j, \overline{z}_k)$ is smaller than a pre-specified parameter θ, then we assign the point $\overline{x}_j$ to k th cluster. Here $d_e(\overline{x}_j, \overline{z}_k)$ is the Euclidean distance between the point $\overline{x}_j$ and the cluster center $\overline{z}_k$. But if $(d_{ps}(\overline{x}_j, \overline{z}_k)/d_e(\overline{x}_j, \overline{z}_k)) > \theta$, assignment is done based on the minimum Euclidean distance criterion as normally used in [6] or the K-means algorithm, i.e., assign $\overline{x}_j$ to k th cluster where $k = Argmin_{i=1,\ldots,K} d_e(\overline{x}_j, \overline{z}_i)$. The reason for doing such an assignment is as follows: in the intermediate stages of the algorithm, when the centers are not yet properly evolved, then the minimum d_{ps} value for a point is expected to be quite large, since the point might not be symmetric with respect to any center. In such cases, using Euclidean distance for cluster assignment appears to be intuitively more appropriate.

The value of θ is kept equal to the maximum nearest neighbor distance among all the points in the data set as done in [3]. It may be noted that if a point is indeed symmetric with respect to some cluster center then the symmetrical distance computed in the above way will be small, and can be bounded as follows. Let d_{NN}^{max} be the maximum nearest neighbor distance in the data set. That is $d_{NN}^{max} = max_{i=1,\ldots,N} d_{NN}(\overline{x}_i)$, where $d_{NN}(\overline{x}_i)$ is the nearest neighbor distance of $\overline{x}_i$. Assuming that reflected point of $\overline{x}$ with respect to the cluster center $\overline{c}$ lies within the data space, it may be noted that $d_1 \leq (d_{NN}^{max}/2)$ and $d_2 \leq (3 \times d_{NN}^{max}/2)$ resulting in $((d_1 + d_2)/2) \leq d_{NN}^{max}$. Ideally, a point $\overline{x}$ is exactly symmetrical with respect to some $\overline{c}$ if $d_1 = 0$. However considering the uncertainty of the location of a point as the sphere of radius d_{NN}^{max} around $\overline{x}$, we have kept the threshold θ equal to d_{NN}^{max}.Thus the computation of θ is automatic and does not require user intervention.

After the assignments are done, the cluster centers encoded in a string are replaced by the mean points of the respective clusters. This is referred to as the K-means like update center operation.

Procedure: clustering_PS()

- **Assignment of data points**:
1. Let a particular string encode total K number of clusters. For all data points $\overline{x}_i$, $1 \leq i \leq n$, compute
$k^* = Argmin_{k=1\ldots K} d_{ps}(\overline{x}_i, \overline{c}_k)$
2.If $(d_{ps}(\overline{x}_i, \overline{c}_{k^*})/d_e(\overline{x}_i, \overline{c}_k) < \theta)$
/*$d_e(\overline{x}_i, \overline{c}_k)$ is the Euclidean distance between the point $\overline{x}_i$ and cluster centroid $\overline{c}_k$*/
assign the data point $\overline{x}_i$ to the k^*th cluster.
3.Otherwise, the data point is assigned to the k^* cluster where
$k^* = Argmin_{k=1\ldots K} d_e(\overline{x}, \overline{c_k})$
- **Updation of centres**: Compute the new centroids of the K clusters as follows:

$$\overline{c}_k(t+1) = \frac{\sum_{i \in S_k(t)} \overline{x_i}}{N_k} \quad (4)$$

where $k = 1, \ldots K$ and $S_k(t)$ is the set of elements that are assigned to the kth cluster at generation t and $N_k = |S_k|$.

Fig. 3. Main steps of clustering_PS() procedure.

4.3. Fitness computation

In this article the well-known Euclidean distance based Xie-Beni (XB) index [14] and the point symmetry distance based *Sym*-index [11,15] are taken as the two objectives that need to be simultaneously optimized. Note that many other cluster validity indices could have been used. It has been already established in [11] that *Sym*-index is more useful to detect symmetrical shaped clusters from data sets. Based on this observation, *Sym*-index is used as one of the cluster validity index to be optimized by AMOSA. XB-index is an Euclidean distance based index. It is a ratio of within cluster compactness (measured using Euclidean distance) to between cluster separation. Thus it can detect hyperspherical shaped clusters well. Here cluster separation is measured using the minimum distance between any two cluster centers. But in *Sym*-index cluster separation is measured by using maximum distance between any two cluster centers. Thus these two indices check different characteristics of the clusters. Moreover, XB-index is a very popular and well-known cluster validity index based on the Euclidean distance. Thus here we have optimized both XB-index and *Sym*-index simultaneously using MOO.

For computing these measures, the centers encoded in a string are first extracted. Let there be K number of cluster centers encoded in a particular string. Let these be denoted as $\mathbf{Z}=\bar{z}_1,\bar{z}_2,\ldots,\bar{z}_K$.

The XB-index is defined as a function of the ratio of the total variation σ to the minimum separation sep of the clusters. Here σ and sep are written as: $\sigma(\mathbf{Z};\mathbf{X})=\sum_{i=1}^{K}\sum_{k=1}^{n_i} d_e^2(\bar{z}_i,\bar{x}_k^i)$, and $sep(\mathbf{Z})=min_{i\neq j}\{\|\bar{z}_i-\bar{z}_j\|^2\}$, where $\|.\|$ is the Euclidean norm, and $d_e(\bar{z}_i,\bar{x}_k^i)$ is the Euclidean distance between the k th point of the i th cluster, $\bar{x}_k^i$, and the cluster center $\bar{z}_i$, and n_i denotes the number of points present in the i th cluster. $\mathbf{Z}$ and $\mathbf{X}$ represent the set of cluster centers and the data set, respectively. The XB-index is then written as

$$XB=\frac{\sigma(\mathbf{Z};\mathbf{X})}{sep(\mathbf{Z})}=\frac{\sum_{i=1}^{K}(\sum_{k=1}^{n_i} d_e^2(\bar{z}_i,\bar{x}_k^i))}{n(min_{i\neq j}(\|\bar{z}_i-\bar{z}_j\|^2))}.$$

Note that when the partitioning is compact and good, the total deviation (σ) should be low while the minimal separation (sep) between any two cluster centers should be high. Thus, the objective is therefore to minimize the XB-index for achieving the proper clustering.

The second objective function is a newly defined point symmetry distance based *Sym*-index [11,15]. It is defined as follows:

$$Sym(K)=\left(\frac{1}{K}\times\frac{1}{\mathscr{E}_K}\times D_K\right), \tag{3}$$

where K is the number of clusters present in that string. Here, $\mathscr{E}_K=\sum_{i=1}^{K}E_i$ such that $E_i=\sum_{j=1}^{n_i} d_{ps}(\bar{x}_j^i,\bar{z}_i)$ and $D_K=max_{i,j=1}^{K}\|\bar{z}_i-\bar{z}_j\|$. D_K is the maximum Euclidean distance between two cluster centers among all centers. $d_{ps}(\bar{x}_j^i,\bar{z}_i)$ is the point symmetry based distance [3] between the j th point of the i th cluster, $\bar{x}_j^i$, and the cluster center $\bar{z}_i$ and n_i is the total number of points in the i th cluster. Here, the first *knear* nearest neighbors of $\bar{x}_j^*=2\times\bar{z}_i-\bar{x}_j^i$ will be searched among only those points which are in cluster i, i.e., the *knear* nearest neighbors of $\bar{x}_j^*$, the reflected point of $\bar{x}_j^i$ with respect to $\bar{z}_i$, and $\bar{x}_j^i$ should belong to the i th cluster.

The objective is to maximize the *Sym*-index in order to obtain the actual number of clusters and to achieve proper clustering. As formulated in Eq. (3), *Sym* is a composition of three factors, these are $1/K$, $1/\mathscr{E}_K$ and D_K. The first factor increases as K decreases; as *Sym* needs to be maximized for optimal clustering, so it will prefer to decrease the value of K. The second factor is the within cluster total symmetrical distance. For cluster which has good symmetrical structure, E_i value is less. This, in turn, indicates that the formation of more number of clusters, which are symmetrical in shape, would be encouraged. Finally the third factor, D_K, measuring the maximum separation between a pair of clusters, increases with the value of K. As these three factors are complementary in nature, so they are expected to compete and balance each other critically for determining the proper partitioning.

4.4. Mutation operation

A new string is generated from the current one by adopting one of the following three types of mutations. Here, at a time, any of the three types of mutation is selected based on the equal probability and applied to the particular string.

(1) Each cluster center encoded in a string is replaced with a random variable drawn from a Laplacian distribution, $p(\varepsilon)\propto e^{-|\varepsilon-\mu|/\delta}$, where the scaling factor δ sets the magnitude of perturbation. Here μ is the value at the position which is to be perturbed. The scaling factor δ is chosen equal to 1.0. The old value at the position is replaced with the newly generated value. Here this type of mutation operator is applied for all dimensions independently.
(2) One randomly generated cluster center is removed from the string, i.e., the total number of clusters encoded in the string is decreased by 1.
(3) The total number of clusters encoded in the string is increased by 1. One randomly chosen point from the data set is encoded as the new cluster center.

Any one of the above mentioned types of mutation is applied randomly on a particular string if it is selected for mutation.

4.5. Termination criteria

The steps of AMOSA based clustering technique (VAMOSA) follow those of the conventional simulated annealing (SA). As like SA, initially in AMOSA temperature is set equal to T_{max}, maximum value of temperature. The steps of AMOSA are executed for total *iter* times per temperature. Temperature is reduced by some cooling rate, α after the execution of total *iter* iterations. The steps of AMOSA are executed until $T>T_{min}$ where T_{min} is the minimum value of temperature.

4.6. Selection of the best solution

In MO, the algorithms produce a large number of non-dominated solutions [2] on the final Pareto optimal front. Each of these solutions provides a way of clustering the given data set. All the solutions are equally important from the algorithmic point of view. But sometimes the user may want only a single solution. Consequently, in this paper a method of selecting a single solution from the set of solutions is now developed. This method is a semi-supervised one.

Here we assume that the class level of 10% of the whole data set (denoted as *test patterns*) is known to us. The proposed VAMOSA algorithm is executed on the rest 90% of the data sets for which no class information is known beforehand. A set of Pareto optimal solutions will be generated. For each clustering associated with a solution from the final Pareto optimal set, the *test patterns* are also assigned cluster labels based on the nearest center criterion, and the amount of misclassification is calculated by computing the *Minkowski score* values. *Minkowski score* is a measure of the quality of a solution given the true clustering [26]. Let T be the "true" solution and S the solution we wish to measure. Denote by n_{11} the number of pairs of elements that are in the same cluster in both S and T. Denote by n_{01} the number of pairs that are in the same cluster only in S, and by n_{10} the number of pairs that are in the same cluster in T. *Minkowski score* is then defined as

$$D_M(T,S)=\sqrt{\frac{n_{01}+n_{10}}{n_{11}+n_{10}}} \tag{5}$$

In this case the optimum score is 0, with lower scores being "better".

The solution with the minimum *Minkowski score* value calculated over the test set is selected as the best solution.

5. Data sets used for experiment

Thirteen data sets are used for the experiment: seven of them are artificial data (*AD_5_2*, *AD_10_2*, *Mixed_5_2*, *Sym_3_2*, *Square1*, *Square4*, *Sizes5*) and six are real-life data sets (*Iris, BreastCancer, Newthyroid, LungCancer, Wine* and *LiverDisorder*).

(1) *AD_5_2*: This data set, used in [6], consists of 250 two-dimensional data points distributed over five spherically shaped clusters. The clusters present in this data set are highly overlapping, each consisting of 50 data points. This data set is shown in Fig. 4(a).
(2) *AD_10_2*: This data set, used in Ref. [27], consists of 500 two-dimensional data points distributed over 10 different clusters. Some clusters are overlapping in nature. Each cluster consists of 50 data points. This data set is shown in Fig. 4(b).
(3) *Mixed_5_2*: This data set, used in Ref. [11], contains 850 data points distributed over five clusters, as shown in Fig. 4(c).
(4) *Sym_3_2*: This data set, used in [3], is a combination of ring-shaped, spherically compact and linear clusters. The total number of points in it is 350. The dimension of this data set is two. This data set is shown in Fig. 5(a).
(5) *Square1*: This data set, used in Ref. [1], consists of 1000 data points distributed over four squared clusters. This is shown in Fig. 5(b).
(6) *Square4*: This data set, used in Ref. [1], consists of 1000 data points distributed over four squared clusters. This is shown in Fig. 6(a).
(7) *Sizes5*: This data set, used in Ref. [1], consists of 1000 data points distributed over four clusters. The densities of these clusters are not uniform. This is shown in Fig. 6(b).
(8) *Iris*: This data set consists of150 four-dimensional data points distributed over three clusters. Each cluster consists of 50 points. This data set represents different categories of irises characterized by four feature values [28]. It has three classes Setosa, Versicolor and Virginica. It is known that two classes (Versicolor and Virginica) have a large amount of overlap while the class Setosa is linearly separable from the other two.
(9) *BreastCancer*: This *Wisconsin breast cancer* data set consists of 683 sample points. Each pattern has nine features corresponding to clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. There are two categories in the data: malignant and benign. The two classes are known to be linearly separable.
(10) *Newthyroid*: The original database from where it has been collected is titled as thyroid gland data ("normal", "hypo" and "hyper" functioning). Five laboratory tests are used to predict whether a patient's thyroid belongs to the class euthyroidism,

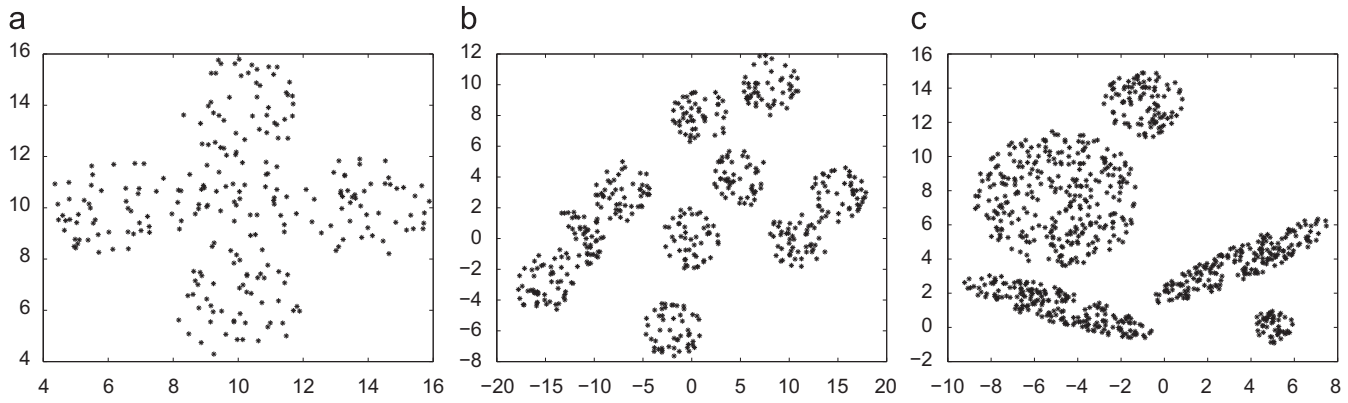

Fig. 4. (a) *AD_5_2*, (b) *AD_10_2* and (c) *Mixed_5_2*.

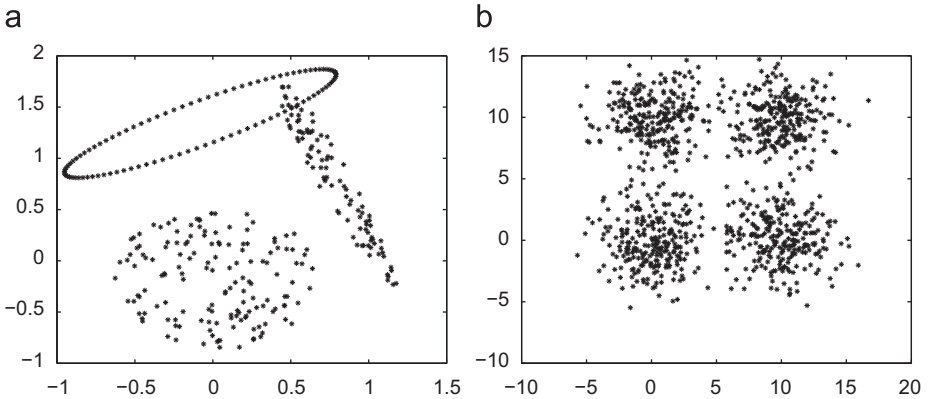

Fig. 5. (a) *Sym_3_2* and (b) *Square1*.

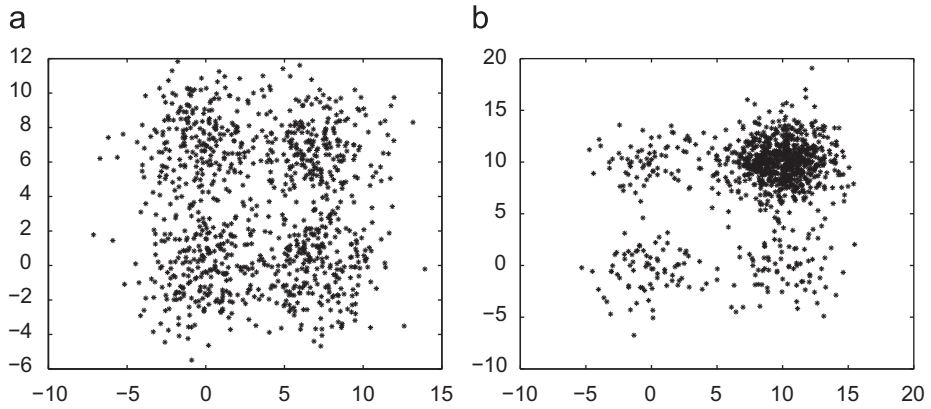

Fig. 6. (a) *Square4* and (b) *Sizes5*.

hypothyroidism or hyperthyroidism. There are a total of 215 instances and the number of attributes is five.

(11) *LungCancer*: These data consist of 32 instances having 56 features each. The data describe three types of pathological lung cancers.

(12) *Wine*: This is the Wine recognition data consisting of 178 instances having 13 features resulting from a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

(13) *LiverDisorder*: This is the liver disorder data consisting of 345 instances having six features each. The data have two categories.

6. Experimental results

The parameters of the proposed VAMOSA clustering technique are as follows: $Tmax=100$, $Tmin=0.00001$, $\alpha=0.8$, $SL=200$ and $HL=100$. Here K^{max} is set equal to $\sqrt{n}$, where n is the size of the data set. The proposed AMOSA based automatic multiobjective clustering technique produces a large number of non-dominated solutions on the final Pareto optimal front. The best solution is identified by the method proposed in Section 4.6. For the purpose of comparison, another MO clustering technique, MOCK [1], is also executed on the above mentioned data sets with default parameter settings. The source code for MOCK is obtained from (http://dbkgroup.org/handl/mock/). In MOCK, the best solution from the final Pareto optimal front is selected by GAP-statistics [29]. Note that for every data set used here for experiment, class labels of all the data points are available. Thus in order to quantify the obtained partitionings by different algorithms, their corresponding *Minkowski score* s (MS) [26] are computed. The number of clusters identified by the best solution of the proposed VAMOSA clustering technique and MOCK clustering technique, and the *Minkowski score* (MS) values of the corresponding partitionings for all the data sets used here for experiment are reported in Table 1.

In order to show the efficacy of the proposed MO clustering technique over existing single objective clustering techniques, two recently developed genetic algorithm based automatic clustering techniques, genetic clustering for unknown K (GCUK clustering) [6] and variable string length genetic point symmetry based clustering technique (VGAPS clustering) [11], are also executed on the above mentioned 13 data sets. These single objective automatic clustering techniques provide a single solution after their execution. GCUK clustering technique optimizes an Euclidean distance based cluster validity index, XB-index [14], by using the search capability of genetic algorithms to automatically determine the appropriate partitioning from data sets. The parameters of the GCUK clustering technique are as follows: population size = 100, number of generations = 40, probability of mutation=0.2 and probability of crossover=0.8. VGAPS clustering technique optimizes a newly developed point symmetry based cluster validity index, *Sym*-index [15], by using the search capability of genetic algorithms. The parameters of the VGAPS clustering technique are as follows: population size = 100, number of generations = 40. The probabilities of mutation and crossover are determined adaptively as in [30]. Here also K is varied between 2 and $\sqrt{n}$. Increasing the number of generations did not improve the performance of the algorithm. The number of clusters automatically determined by these clustering techniques for the 13 data sets is also reported in Table 1. The MS values are also calculated for the partitionings obtained by these two single objective clustering techniques for these 13 data sets. These are also reported in Table 1.

6.1. Results on artificial data sets

(1) *AD_5_2*: As can be seen from Table 1, for this data set VGAPS and the proposed VAMOSA clustering techniques perform similarly. The corresponding partitioning is shown in Fig. 7(a). The best solution provided by MOCK is not able to determine the appropriate number of clusters from this data set. The corresponding partitioning is shown in Fig. 7(b). GCUK clustering optimizing XB-index is able to detect the appropriate number of clusters from this data set and the corresponding partitioning is very near to the actual partitioning of the data set (refer to Table 1). The corresponding partitioning is shown in Fig. 7(c).

(2) *AD_10_2*: For this data set, GCUK clustering provides the best partitioning (shown in Fig. 9(b)) and the corresponding MS value is also the minimum (refer to Table 1). VAMOSA clustering technique is also able to detect the appropriate number of clusters from this data set but the corresponding MS value is slightly higher than that of GCUK clustering (refer to Table 1). The corresponding partitioning is shown in Fig. 8(a). But both MOCK and VGAPS clustering techniques are not able to detect the appropriate number of clusters from this data set. The partitionings identified by MOCK and VGAPS clustering techniques for this data set are shown in Figs. 8 and 9, respectively.

(3) *Mixed_5_2*: For this data set, the three clustering techniques, proposed *VAMOSA*, VGAPS and MOCK, are able to detect the appropriate number of clusters and the appropriate partitioning (refer to Table 1). The corresponding partitioning is shown in Fig. 10(a). GCUK clustering optimizing XB-index is not able to detect the appropriate partitioning from this data set. It overestimates the number of clusters from these data (refer to Table 1). Here it breaks some larger clusters into several smaller clusters.

Table 1
Number of clusters and the *Minkowski score* (MS) values obtained by VAMOSA clustering technique, another automatic MO clustering technique, MOCK, two single objective clustering techniques, VGAPS clustering optimizing *Sym*-index and GCUK clustering optimizing XB-index, for all the data sets used here for experiment.

DataSet	AC	VAMOSA		MOCK		VGAPS		GCUK	
		OC	MS	OC	MS	OC	MS	OC	MS
AD_5_2	5	**5**	**0.25**	6	0.39	**5**	**0.25**	5	0.39
AD_10_2	10	10	0.43	6	1.01	7	0.84	**10**	**0.09**
Mixed_5_2	5	**5**	**0.00**	**5**	**0.00**	**5**	**0.00**	9	0.75
Sym_3_2	3	**3**	0.12	2	0.69	**3**	0.12	4	0.74
Square1	4	4	0.19	**4**	0.19	4	0.20	4	0.21
Square4	4	**4**	**0.51**	4	0.60	5	0.52	**4**	**0.51**
Sizes5	4	**4**	**0.14**	2	0.64	5	0.22	4	0.25
Iris	3	2	0.80	2	0.82	**3**	**0.62**	2	0.84
Cancer	2	**2**	**0.32**	2	0.39	2	0.37	2	0.38
Newthyroid	3	5	0.57	2	0.82	3	0.58	6	0.65
Lungcancer	3	**3**	**0.85**	7	0.97	2	0.97	6	0.94
Wine	3	3	0.97	**3**	**0.90**	6	0.97	6	0.93
LiverDisorder	2	**2**	**0.98**	3	0.98	**2**	**0.98**	2	0.99

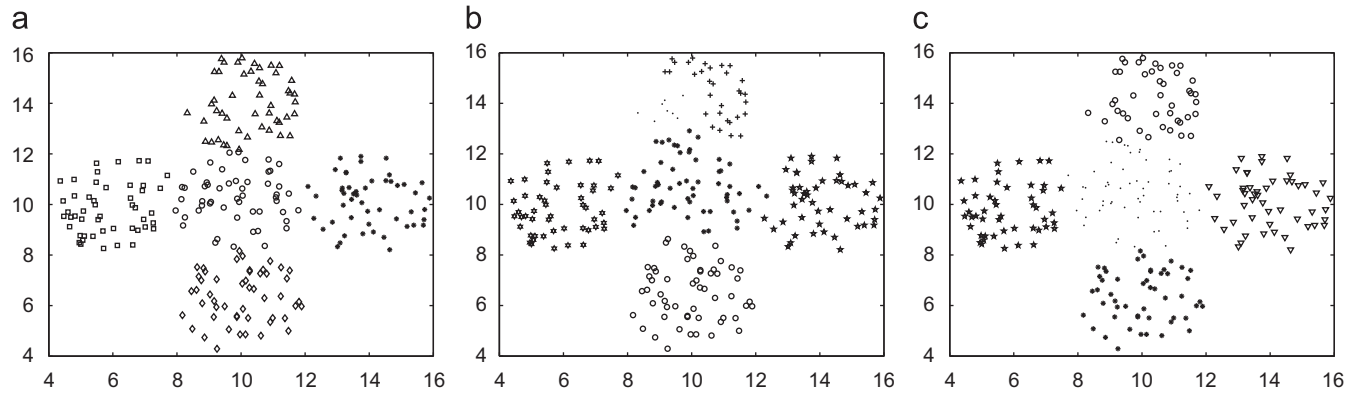

Fig. 7. Automatically clustered *AD_5_2* after application of (a) VAMOSA/VGAPS clustering technique for $K=5$, (b) MOCK clustering technique for $K=6$ and (c) GCUK clustering technique for $K=5$.

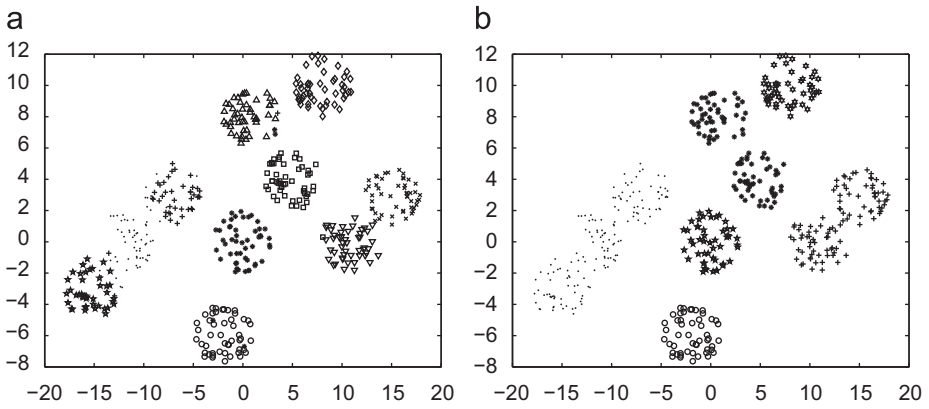

Fig. 8. Automatically clustered *AD_10_2* after application of (a) VAMOSA clustering technique for $K=10$ and (b) MOCK clustering technique for $K=6$.

(4) *Sym_3_2*: As seen from Table 1, both the symmetry based clustering techniques, proposed VAMOSA and VGAPS, are able to detect the proper number of clusters and the proper partitioning from this data set. The corresponding partitioning is shown in Fig. 10(b). MOCK again merges the two overlapping clusters into one cluster and provides $K=2$ as the optimal number of clusters. GCUK clustering technique identifies total $K=4$ number of clusters from this data set. The MS scores reported in Table 1 also show the poorer performance of both MOCK and GCUK clustering techniques for this data set.

(5) *Square1*: All the four clustering algorithms are able to detect the appropriate number of clusters from this data set. The partitioning identified by VAMOSA clustering technique is shown in Fig. 11(a). Table 1 reveals that the MS values of the obtained partitionings by the proposed VAMOSA, VGAPS and MOCK clustering techniques are also the same but GCUK optimizing XB-index attains a slightly higher MS score.

(6) *Square4*: For this data set, except VGAPS, all the three clustering algorithms are able to detect the appropriate number of clusters (refer to Table 1). VGAPS provides $K=5$ as the optimal number

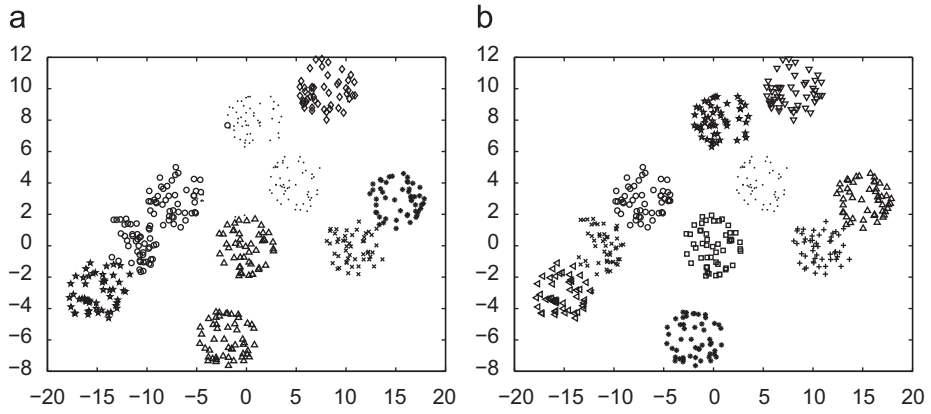

Fig. 9. Automatically clustered *AD_10_2* after application of (a) VGAPS clustering technique for $K=7$ and (b) GCUK clustering technique for $K=10$.

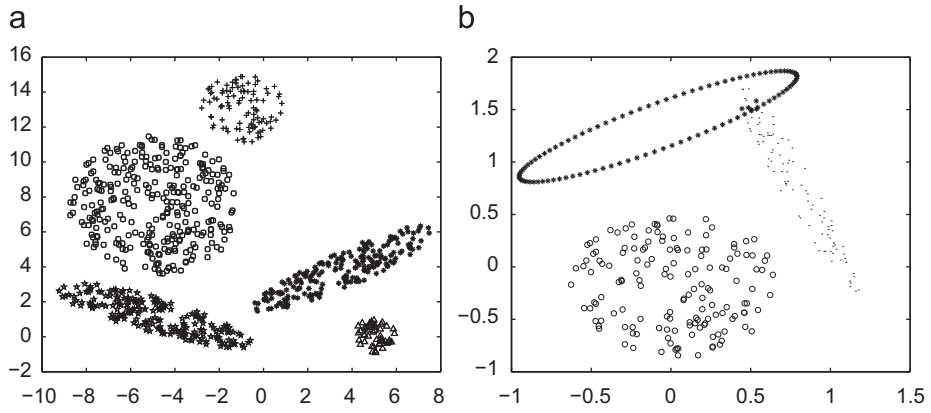

Fig. 10. (a) Automatically clustered *Mixed_5_2* after the application of VAMOSA/MOCK/VGAPS clustering technique for $K=5$, (b) automatically clustered *Sym_3_2* after the application of VAMOSA/VGAPS clustering technique for $K=3$.

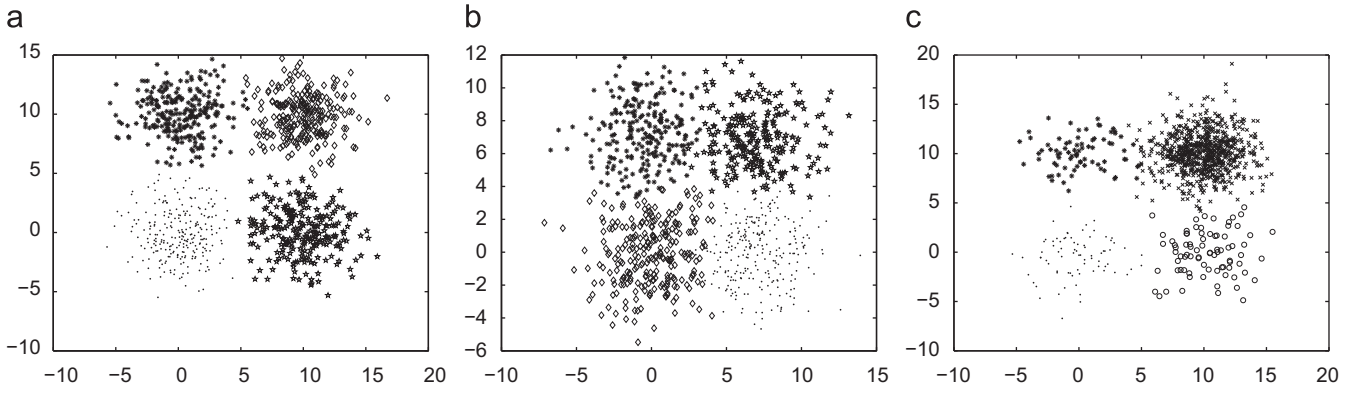

Fig. 11. (a) Automatically clustered *Square1* after application of VAMOSA clustering technique for $K=4$, (b) automatically clustered *Square1* after application of VAMOSA clustering technique for $K=4$ and (c) automatically clustered *Sizes5* after application of VAMOSA clustering technique for $K=4$.

of clusters. The partitioning identified by VAMOSA clustering technique is shown in Fig. 11(b). Table 1 reveals that the MS values of the obtained partitionings by the proposed VAMOSA and GCUK clustering techniques are same. Although MOCK is able to detect the correct number of clusters but the corresponding MS value is the highest among all these four clustering algorithms.

(7) *Sizes5*: Here only the proposed VAMOSA clustering technique and the GCUK clustering technique are able to detect the appropriate number of clusters. But the MS value corresponding to the partitioning obtained by VAMOSA is much lesser than that of GCUK (refer to Table 1). The partitioning identified by VAMOSA clustering technique is shown in Fig. 11(c). The best solution of MOCK provides $K=2$ as the optimal number of clusters. VGAPS overestimates the number of clusters from this data set. It breaks the maximum densed cluster into two clusters.

6.2. Results on real-life data sets

(1) *Iris*: For this real-life data set, only VGAPS clustering technique is able to determine the appropriate number of clusters. The corresponding MS score is also the minimum (refer to

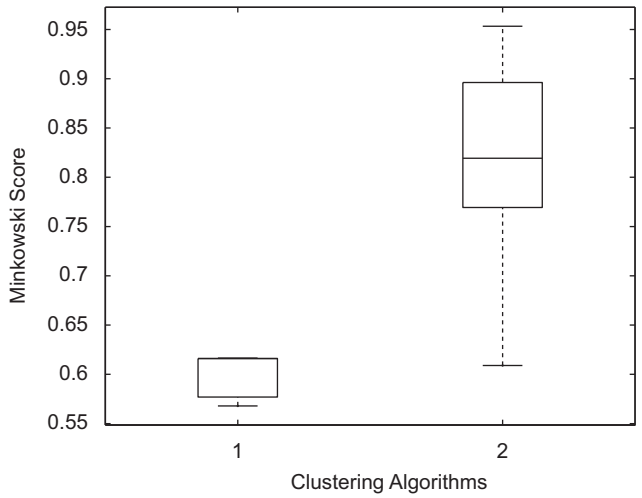

Fig. 12. Boxplots of the Minkowski scores of the Pareto optimal solutions obtained by VAMOSA clustering technique and MOCK clustering technique for *Newthyroid* data set. Here column "1" denotes the VAMOSA clustering technique and column "2" denotes the MOCK clustering technique.

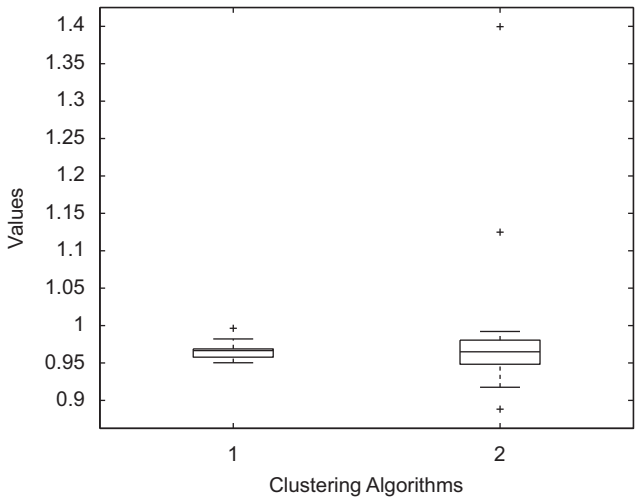

Fig. 13. Boxplots of the Minkowski scores of the Pareto optimal solutions obtained by VAMOSA clustering technique and MOCK clustering technique for *Wine* data set. Here column "1" denotes the VAMOSA clustering technique and column "2" denotes the MOCK clustering technique.

Table 1). As this is a higher dimensional data set, no visualization is possible. Other three clustering algorithms, newly proposed VAMOSA, MOCK and GCUK, provide $K=2$ as the optimal number of clusters, which is also often obtained for many other methods for *Iris* data set. This is because in *Iris* data set, the two clusters are highly overlapping to each other whereas the third cluster is well separated from these two. Thus many methods provide $K=2$ as the optimal number of clusters. But the MS score corresponding to VAMOSA for $K=2$ is the minimum among these three clustering techniques (refer to Table 1).

(2) *Cancer*: For this data set all the four clustering techniques are able to detect the appropriate number of clusters ($K=2$ for this case). But the MS value obtained by VAMOSA clustering technique is the minimum (refer to Table 1).

(3) *Newthyroid*: For this real-life data set only the VGAPS clustering technique is able to detect the appropriate number of clusters ($K=3$ in this case). VAMOSA provides $K=5$ as the optimal number of clusters. But the MS value obtained by the final solution provided by VAMOSA clustering technique is lesser than that obtained by VGAPS (refer to Table 1). Both MOCK and GCUK clustering techniques are not able to determine the appropriate number of clusters from this data set. MOCK attains the highest MS value compared to all other three algorithms. For the purpose of comparison the final Pareto optimal front obtained by VAMOSA clustering technique is shown in Fig. 15(a). The boxplots of the *Minkowski score* values of the solutions on the final Pareto optimal front provided by VAMOSA and MOCK clustering techniques are shown in Fig. 12 for the purpose of illustration. This figure reveals that the MS values over the final Pareto optimal front provided by VAMOSA are much less than those of MOCK clustering technique.

(4) *LungCancer*: For this high dimensional data set only the proposed VAMOSA clustering technique is able to detect the appropriate number of clusters. None of the other algorithms are able to detect the correct number of clusters. The MS value obtained by VAMOSA is also the minimum (refer to Table 1). The final Pareto optimal front obtained by the proposed VAMOSA clustering technique is shown in Fig. 15(b).

(5) *Wine*: For this real-life data set both the MOO clustering techniques, VAMOSA and MOCK, are able to determine the appropriate number of clusters. But the MS value attained by MOCK is lesser than that of VAMOSA (refer to Table 1). This may be due to the absence of symmetrical shaped clusters in this data set. Both VGAPS and GCUK clustering techniques provide $K=6$ as the optimal number of clusters. The final Pareto optimal front obtained by VAMOSA clustering technique is shown in Fig. 16(a). The boxplots of the *Minkowski score* values of the solutions on the final Pareto optimal front provided by VAMOSA and MOCK clustering techniques are shown in Fig. 13 for the purpose of illustration.

(6) *LiverDisorder*: For this data set only VAMOSA and GCUK clustering techniques are able to detect the appropriate number of clusters. But the corresponding MS values are quite high. This may be due to the mismatch in the clustering structures and the class levels present in this data set. Other two algorithms, MOCK and VGAPS, provide $K=3$ as the proper number of clusters. The final Pareto optimal front obtained by VAMOSA clustering technique is shown in Fig. 16(b). The boxplots of the *Minkowski score* values of the solutions on the final Pareto optimal front provided by VAMOSA and MOCK clustering techniques are shown in Fig. 14 for the purpose of illustration.

6.3. Summary of results

It can be seen from the above results that proposed VAMOSA clustering technique is able to detect the appropriate partitioning and the appropriate number of clusters from most of the data sets used here for experiment. It outperforms another MO clustering technique, MOCK, and two single objective genetic algorithm based clustering techniques. The superiority of VAMOSA is also established on six real-life data sets. These real-life data sets are of different characteristics with the number of dimensions varying from 4 to 56. Results on 13 artificial and real-life data sets establish the fact that VAMOSA is well suited to detect the number of clusters automatically from data sets having clusters of widely varying characteristics as long as they possess the property of point symmetry. MOCK is generally able to detect well separated/hyperspherical shaped clusters. But it fails for overlapping symmetrical shaped clusters. Results also show that VGAPS is only able to detect symmetrical shaped clusters well. GCUK is capable to handle only hyperspherical shaped clusters.

6.4. Statistical test

Here we have done some statistical tests guided by [17,18] to establish the superiority of the proposed clustering technique, VAMOSA. We have done Friedman statistical test [31] to detect whether the four clustering techniques used here for experiment perform similarly or not. It assigns ranks to each algorithm for each data set. It tests whether the measured average ranks are significantly different from the mean rank. Friedman test shows that measured average ranks and mean rank are different with a p value of 0.0026. The corresponding table is shown in Table 2. Finally Nemenyi's test [32] is performed to compare the clustering techniques pairwise. In each case $\alpha = 0.05$. These results are also reported in Table 3. Note that in most of the cases except while comparing MOCK and GCUK clustering techniques and, VGAPS and GCUK clustering techniques, the null hypotheses (the pairing algorithms perform similarly) are rejected as the corresponding p values are smaller than the αs.

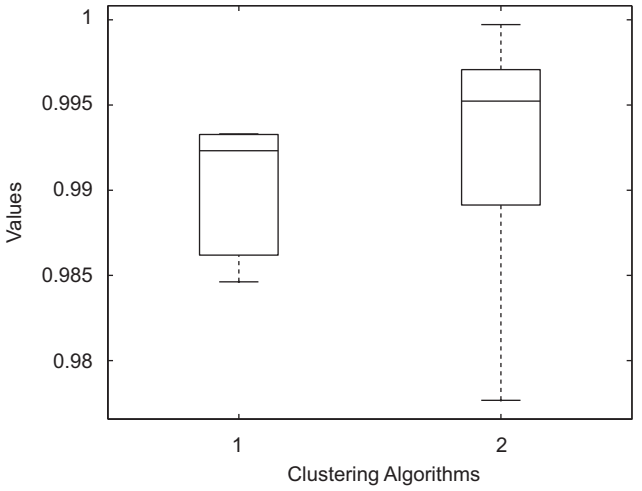

Fig. 14. Boxplots of the Minkowski scores of the Pareto optimal solutions obtained by VAMOSA clustering technique and MOCK clustering technique for *LiverDisorder* data set. Here column "1" denotes the *VAMOSA* clustering technique and column "2" denotes the MOCK clustering technique.

6.5. Exploring the objective functions used

Here, we have established the effectiveness of using the XB-index along with *Sym*-index in VAMOSA clustering technique as two objectives vis-a-vis some other cluster validity indices, two Euclidean distance based cluster validity indices, PBM-index [33] and DB-index [34]. The number of clusters obtained after applying VAMOSA optimizing three different sets of objective functions, VAMOSA(A) optimizing XB-index and *Sym*-index simultaneously, VAMOSA(B) optimizing DB-index and *Sym*-index simultaneously, VAMOSA(C) optimizing PBM-index and *Sym*-index simultaneously, for all the data sets is shown in Table 4. This table shows that for nine, eight and four data sets out of 13 data sets VAMOSA(A), VAMOSA(B) and VAMOSA(C) clustering techniques perform the best, respectively. Thus it can be concluded that VAMOSA clustering with

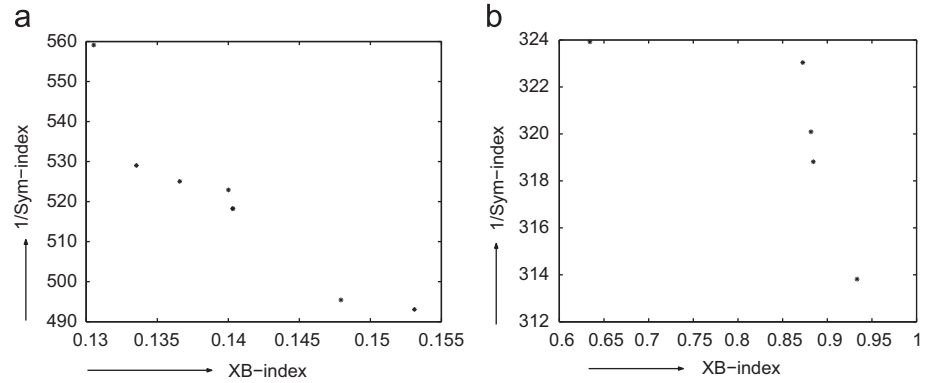

Fig. 15. Pareto optimal front obtained by the proposed VAMOSA clustering technique for (a) *Newthyroid* data set and (b) *Lungcancer* data set.

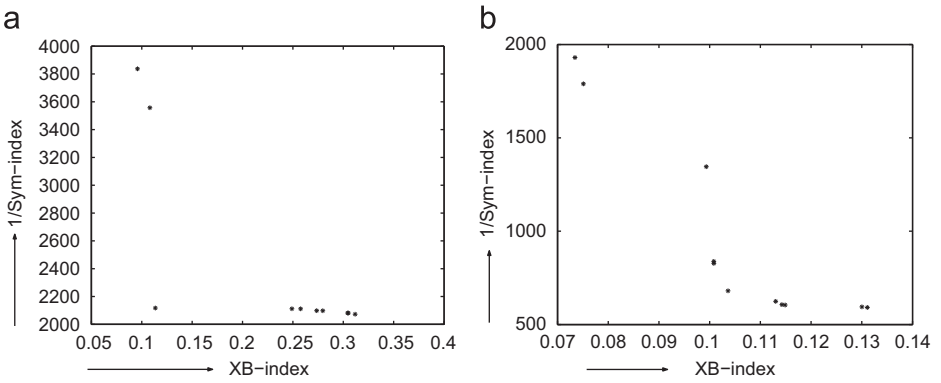

Fig. 16. Pareto optimal front obtained by the proposed *VAMOSA* clustering technique for (a) *Wine* data set and (b) *LiverDisorder* data set.

Table 2
Computation of the rankings for the four algorithms considered in the study over 13 data sets, based on the MS value obtained.

DataSet	VAMOSA	MOCK	VGAPS	GCUK
AD_5_2	0.25(1)	0.39(2)	0.25(1)	0.39(2)
AD_10_2	0.43(2)	1.01(4)	0.84(3)	0.09(1)
Mixed_5_2	0.00(1)	0.00(1)	0.00(1)	0.75(2)
Sym_3_2	0.12(1)	0.69(2)	0.12(1)	0.74(3)
Square1	0.19(1)	0.19(1)	0.20(2)	0.21(3)
Square4	0.51(1)	0.60(3)	0.52(2)	0.51(1)
Sizes5	0.14(1)	0.64(4)	0.22(2)	0.25(3)
Iris	0.80(2)	0.82(3)	0.62(1)	0.84(4)
Cancer	0.32(1)	0.39(4)	0.37(2)	0.38(3)
Newthyroid	0.57(1)	0.82(4)	0.58(2)	0.65(3)
Lungcancer	0.85(1)	0.97(3)	0.97(3)	0.94(2)
Wine	0.97(3)	0.90(1)	0.97(3)	0.93(2)
LiverDisorder	0.98(1)	0.98(1)	0.98(1)	0.99(2)
Average rank	1.31	2.538	1.846	2.384

Table 3
Results of Nemenyi's test for different pairs of clustering algorithms.

Clustering algo.	Pairing clustering algo.	*p*-Value
VAMOSA	MOCK	0.0067
VAMOSA	VGAPS	0.0339
VAMOSA	GCUK	0.0075
MOCK	VGAPS	0.0348
MOCK	GCUK	1
VGAPS	GCUK	0.0707

XB-index and *Sym*-index as objective functions performs the best compared to other two versions.

6.6. Results using different parameter settings

The annealing schedule of an SA algorithm consists of (i) initial value of temperature (*Tmax*), (ii) minimum value of temperature (*Tmin*), (iii) cooling schedule, (iv) number of iterations to be performed at each temperature and (v) stopping criterion to terminate the algorithm. A very short discussion regarding the proper choice of these parameters has been described in Section IV.D of the paper on AMOSA [12]. It indeed depends on the data sets used. Here these parameters are selected manually.

In this paper we have shown the sensitivity of the results of VAMOSA on two data sets, *AD_10_2* and *Iris*, for different values of *Tmax*, *Tmin*, *SL* and *HL*. Here seven different parameter settings are used:

(1) Setting 1: $Tmax = 100$, $Tmin = 0.00001$, $SL = 200$, $HL = 100$.
(2) Setting 2: $Tmax = 10$, $Tmin = 0.01$, $SL = 30, HL = 20$.
(3) Setting 3: $Tmax = 10$, $Tmin = 0.1$, $SL = 50$, $HL = 40$.
(4) Setting 4: $Tmax = 10$, $Tmin = 0.01$, $SL = 200$, $HL = 100$.
(5) Setting 5: $Tmax = 10$, $Tmin = 0.1$, $SL = 200$, $HL = 100$.
(6) Setting 6: $Tmax = 100$, $Tmin = 0.00001$, $SL = 50$, $HL = 40$.
(7) Setting 7: $Tmax = 100$, $Tmin = 0.00001$, $SL = 30$, $HL = 20$.

The number of clusters and the *Minkowski score* (MS) [26] values obtained by VAMOSA clustering technique with these seven different parameter settings for two data sets are shown in Table 5. Results show that for reasonably good performance of VAMOSA, either *SL*/*HL* value should be high or there should be sufficient number of iterations (i.e., difference between *Tmax* and *Tmin* should be high enough). It can also be concluded that the parameter settings (Setting 1) used in this paper are applicable for almost all data sets.

Table 4
Number of clusters and the *Minkowski score* (MS) values obtained by VAMOSA clustering technique optimizing simultaneously (A) XB-index and *Sym*-index, (B) DB-index and *Sym*-index, (C) PBM-index and *Sym*-index for all the data sets used here for experiment.

DataSet	AC	VAMOSA (A)		VAMOSA (B)		VAMOSA (C)	
		OC	MS	OC	MS	OC	MS
AD_5_2	5	**5**	**0.25**	5	0.65	6	0.64
AD_10_2	10	10	0.43	10	0.47	**10**	**0.37**
Mixed_5_2	5	**5**	**0.00**	**5**	**0.00**	**5**	**0.00**
Sym_3_2	3	**3**	**0.12**	**3**	**0.12**	**3**	**0.12**
Square1	4	**4**	**0.19**	**4**	**0.19**	5	0.27
Square4	4	4	0.51	**4**	**0.49**	4	0.52
Sizes5	4	**4**	**0.14**	4	0.17	6	0.22
Iris	3	2	**0.80**	3	**0.80**	3	**0.80**
Cancer	2	**2**	**0.32**	**2**	**0.32**	3	0.37
Newthyroid	3	5	0.57	8	**0.54**	7	0.56
Lungcancer	3	**3**	**0.85**	2	1.46	2	1.45
Wine	3	3	0.97	9	**0.94**	14	0.96
LiverDisorder	2	**2**	**0.98**	4	0.99	3	0.98

Table 5
Number of clusters and the *Minkowski score* (MS) values obtained by VAMOSA clustering technique for two data sets with seven different parameter settings.

Parameter setting	*AD_10_2*		*Iris*	
	OC	MS	OC	MS
Setting 1	10	0.43	2	0.80
Setting 2	10	0.45	3	0.80
Setting 3	10	0.45	3	0.80
Setting 4	10	0.44	3	0.80
Setting 5	10	0.44	3	0.80
Setting 6	10	0.44	3	0.80
Setting 7	10	0.44	3	0.80

6.7. Stability of the proposed algorithm

In order to show the stability of the proposed clustering technique, VAMOSA is executed on *AD_10_2* and *Iris* data sets with five different initial seeds. Finally in each case, same number of clusters and MS values are obtained. This shows that the proposed algorithm is stable irrespective of the random initialization of the centers. It converges to the same set of solutions irrespective of the initial seed.

6.8. Runtime

The time complexity of AMOSA is $O(Total\,Iter \times HL \times (M + \log(HL)))$ where *Total Iter* is the total number of iterations, *HL* is the hardlimit and *M* is the number of objectives. Here $M = 2$.

(1) As discussed above Kd-tree data structure has been used in order to find the nearest neighbor of a particular point. The construction of Kd-tree requires $O(N \log N)$ time and $O(N)$ space [24] where *N* is the size of the data set.
(2) Initialization of VAMOSA needs $SL \times stringlength$ time where *stringlength* indicates the length of each string in the VAMOSA. Note that for a maximum of K^{max} clusters in *d*-dimensional space, *stringlength* will become $K^{max} \times d$.
(3) Fitness is computed by executing the *clustering_PS* procedure.
    (a) In order to assign each point to a cluster we have to calculate the minimum symmetrical distance of that point with respect to all clusters. For this purpose the Kd-tree based nearest neighbor search is used. If the points are roughly uniformly distributed, then the expected case complexity is $O(c^d + \log N)$, where *c* is a constant depending on dimension

and the point distribution. This is $O(\log N)$ if the dimension d is a constant [35]. Friedman et al. [36] also reported $O(\log N)$ expected time for finding the nearest neighbor. So in order to find minimal symmetrical distance of a particular point, $O(K^{max} \log N)$ time is needed. For N points total complexity becomes $O(K^{max} N \log N)$.

(b) For updating the centers total complexity is $O(K^{max})$.

(c) For calculating the objective functions total complexity will be $O(N)$.

So the fitness evaluation has total complexity $= O(K^{max} N \log N)$ per iteration.

(4) Mutation requires $O(stringlength) = O(K^{max} d)$ time per iteration.

(5) At the end of VAMOSA, the best solution has to be chosen from the final Pareto optimal front. This requires $O(HL \times N)$ time.

So in general total time complexity (combining all the complexities including those by AMOSA and assuming that $HL < N$ and $M \ll N$) becomes $O(K^{max} \times N \times \log N \times Total\,Iter)$. Thus total complexity of VAMOSA clustering is $O(K^{max} \times N \times \log N \times Total\,Iter)$.

Here VAMOSA is executed on a HP xw8400 Workstation with Dual Core 3.0 GHz Intel Xeon processors, 4 MB Cache memory and having 2 GB primary memory. The time taken by VAMOSA clustering technique for all the data sets used here for experiment is also reported here. For *AD_5_2*, *AD_10_2*, *Mixed_5_2*, *Sym_3_2*, *Square1*, *Square4*, *Sizes5*, *Iris*, *Cancer*, *Newthyroid*, *Lungcancer*, *Wine* and *LiverDisorder* data sets VAMOSA clustering technique takes 2 min 16 s, 4 min 44 s, 7 min 30 s, 3 min 45 s, 8 min 10 s, 8 min 9 s, 8 min 8 s, 1 min 56 s, 12 min 15 s, 2 min 56 s, 58 s, 3 min 10 s and 5 min 40 s, respectively.

7. Conclusion and future work

In this article, a multiobjective clustering technique based on a newly developed point symmetry based distance has been developed which can automatically determine the proper number of clusters and the proper partitioning from a given data set. Here assignment of points to different clusters is done based on the point symmetry based distance rather than the Euclidean distance. Two cluster validity measures, one based on the newly developed point symmetry based distance, *Sym*-index, and another based on the Euclidean distance, *XB*-index, are optimized simultaneously. In this regard, a newly developed multiobjective simulated annealing based technique, AMOSA, has been used in this article. The effectiveness of the proposed clustering technique in detecting the proper number of partitions and the proper partitioning is shown for seven artificial and six real-life data sets and the results are compared with those obtained by another MO clustering technique, MOCK [1], two single objective automatic genetic clustering techniques, GCUK clustering optimizing XB-index [6] and VGAPS clustering [11] optimizing a point symmetry based cluster validity index, *Sym*-index [15]. Note that the proposed clustering technique is able to detect clusters having point symmetry property. However VAMOSA will fail for clusters having non-symmetrical shapes. For these clusters it will break them into several symmetrical shaped clusters.

Much further work is needed to investigate using different and more objectives, and to test the approach still more extensively. Selecting the best solution(s) from the Pareto optimal front is an important problem in multiobjective clustering. One semi-supervised method of selecting a single solution from the Pareto optimal front is proposed here. But this method *a priori* assumes that the labeling of the partial data points is known beforehand. Thus some new unsupervised methods to choose the best solution from the final Pareto optimal front have to be developed. The authors are currently working in this direction. Testing cluster symmetry also constitutes an interesting research direction.

References

[1] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Transactions on Evolutionary Computation 11 (1) (2007) 56–76.

[2] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, England, 2001.

[3] S. Bandyopadhyay, S. Saha, GAPS: a clustering method using a new point symmetry based distance measure, Pattern Recognition 40 (2007) 3430–3451.

[4] S. Bandyopadhyay, U. Maulik, Nonparametric genetic clustering: comparison of validity indices, IEEE Transactions on Systems, Man and Cybernetics 31 (1) (2001) 120–125.

[5] R.H. Eduardo, F.F.E. Nelson, A genetic algorithm for cluster analysis, Intelligent Data Analysis 7 (2003) 15–25.

[6] S. Bandyopadhyay, U. Maulik, Genetic clustering for automatic evolution of clusters and application to image classification, Pattern Recognition 2 (2002) 1197–1208.

[7] J.H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, AnnArbor, 1975.

[8] W. Sheng, S. Swift, L. Zhang, X. Liu, A weighted sum validity function for clustering with a hybrid niching genetic algorithm, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 35 (6) 2005.

[9] S. Bandyopadhyay, U. Maulik, A. Mukhopadhyay, Multiobjective genetic clustering for pixel classification in remote sensing imagery, IEEE Transactions on Geoscience and Remote Sensing 45 (5) (2007) 1506–1511.

[10] A.K. Jain, P. Duin, M. Jianchang, Statistical pattern recognition: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.

[11] S. Bandyopadhyay, S. Saha, A point symmetry based clustering technique for automatic evolution of clusters, IEEE Transactions on Knowledge and Data Engineering 20 (11) (2008) 1–17.

[12] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing based multi-objective optimization algorithm: AMOSA, IEEE Transactions on Evolutionary Computation 12 (3) (2008) 269–283.

[13] Y.J. Park, M.S. Song, A genetic algorithm for clustering problems, in: Proceedings of 3rd Annual Conference on Genetic Programming, 1998, pp. 568–575.

[14] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 841–847.

[15] S. Saha, S. Bandyopadhyay, Application of a new symmetry based cluster validity index for satellite image segmentation, IEEE Geoscience and Remote Sensing Letters 5 (2) (2008) 166–170.

[16] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1650–1654.

[17] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[18] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.

[19] D.V. Veldhuizen, G. Lamont, Multiobjective evolutionary algorithms: analyzing the state-of-the-art, Evolutionary Computation 2 (2000) 125–147.

[20] P.J.M. van Laarhoven, E.H.L. Aarts, Simulated Annealing: Theory and Applications, Kluwer Academic Publisher, Dordrecht, 1987.

[21] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6) (1984) 721–741.

[22] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002).

[23] M.-C. Su, C.-H. Chou, A modified version of the *k*-means algorithm with a distance based on cluster symmetry, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 674–680.

[24] M.R. Anderberg, Computational Geometry: Algorithms and Applications, Springer, Berlin, 2000.

[25] D.M. Mount, S. Arya, ANN: a library for approximate nearest neighbor searching, 2005 ⟨http://www.cs.umd.edu/~mount/ANN⟩.

[26] A. Ben-Hur, I. Guyon, Detecting Stable Clusters Using Principal Component Analysis in Methods in Molecular Biology, Humanae Press, Clifton, UK, 2003.

[27] S. Bandyopadhyay, S.K. Pal, Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence, Springer, Heidelberg, 2007.

[28] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 3 (1936) 179–188.

[29] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters via the gap statistics, Journal of the Royal Statistical Society 16 (2004) 1299–1323.

[30] M. Srinivas, L. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Transactions on Systems, Man and Cybernetics 24 (4) (1994) 656–667.

[31] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association 32 (1937) 675–701.

[32] P.B. Nemenyi, Distribution-free multiple comparisons, Ph.D. Thesis, 1963

[33] M.K. Pakhira, U. Maulik, S. Bandyopadhyay, Validity index for crisp and fuzzy clusters, Pattern Recognition 37 (3) (2004) 487–501.

[34] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (1979) 224–227.

[35] J.L. Bentley, B.W. Weide, A.C. Yao, Optimal expected-time algorithms for closest point problems, ACM Transactions on Mathematical Software 6 (4) (1980) 563–580.

[36] J.H. Friedman, J.L. Bently, R.A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Transactions on Mathematical Software 3 (3) (1977) 209–226.

About the Author—SRIPARNA SAHA received the BTech degree in computer science and engineering from the University of Kalyani, Kalyani, India, in 2003 and the MTech degree in computer science from the Indian Statistical Institute, Kolkata, India, in 2005, where she is currently working toward the PhD degree. She is the recipient of the Lt Rashi Roy Memorial Gold Medal from the Indian Statistical Institute for outstanding performance in MTech (computer science). She has coauthored more than 20 articles in international journals and conference/workshop proceedings. She is the recipient of the Google India Women in Engineering Award, 2008. Her research interests include multiobjective optimization, pattern recognition, evolutionary algorithms, and data mining. She is a Student Member of the IEEE.

About the Author—SANGHAMITRA BANDYOPADHYAY received the BS, MS, and PhD degrees in computer science, in 1991, 1993, and 1998, respectively. She is currently an Professor at the Indian Statistical Institute, Kolkata, India. She has worked at the Los Alamos National Laboratory, Los Alamos, New Mexico, University of New South Wales, Sydney, Australia, University of Texas at Arlington, University of Maryland at Baltimore, Fraunhofer Institute, Germany, and Tsinghua University, China. She is the first recipient of the Dr. Shanker Dayal Sharma Gold Medal and also the Institute Silver Medal for being adjudged the best all-around postgraduate performer in IIT, Kharagpur, India, in 1994. She has also received the Young Scientist Awards of the Indian National Science Academy (INSA) and the Indian Science Congress Association (ISCA) in 2000. In 2002, she received the Young Engineer Award of the Indian National Academy of Engineers (INAE) and the Swarnajayanti Fellowship from the Department of Science and Technology (DST) in 2007. She was an Invited Speaker at the Eighth International Conference on Human and Computers 2005, held in Aizu, Japan, from 30 August to 2 September 2005. She has coauthored more than 125 technical articles in international journals, book chapters, and conference/workshop proceedings. She has delivered many invited talks and tutorials around the world. She was the Program Cochair of the First International Conference on Pattern Recognition and Machine Intelligence (PReMI '05) held in Kolkata, during 18–22 December 2005. She has recently published an authored book titled "Classification and Learning Using Genetic Algorithms: Applications in Bioinformatics and Web Intelligence" from Springer and two edited books titled "Advanced Methods for Knowledge Discovery from Complex Data," published by Springer, United Kingdom, in 2005, and "Analysis of Biological Data: A Soft Computing Approach" published by World Scientific in 2007. She has also edited journals special issues in the area of soft computing, data mining, and bioinformatics. Her research interests include computational biology and bioinformatics, soft and evolutionary computation, image processing, pattern recognition, and data mining. She is a Senior Member of the IEEE.