Soft Computing (2019) 23:5693–5714
https://doi.org/10.1007/s00500-018-3227-5

METHODOLOGIES AND APPLICATION

Bi-clustering of microarray data using a symmetry-based multi-objective optimization framework

Sudipta Acharya[1] · **Sriparna Saha**[1] · **Pracheta Sahoo**[2]

Published online: 9 May 2018

Abstract
High-throughput technologies, like DNA microarray, help in simultaneous monitoring of the expression levels of thousands of genes during important biological processes and over the collection of experimental conditions. Automatically uncovering functionally related genes is a basic building block to solve various problems related to functional genomics. But sometimes a subset of genes may not be similar with respect to all the conditions present in the dataset; thus, bi-clustering concept becomes popular where different subsets of genes and the corresponding subsets of conditions with respect to which genes are most similar are automatically identified. In the current study, we have posed this problem in the multi-objective optimization (MOO) framework where different bi-cluster quality measures are optimized simultaneously. The search potentiality of a simulated annealing-based MOO technique, AMOSA, is used for the simultaneous optimization of these measures. A case study on the suitability of different distance measures in solving the bi-clustering problem is also conducted. The competency of the proposed multi-objective-based bi-clustering approach is shown for three benchmark datasets. The obtained results are further validated using statistical and biological significance tests.

Keywords Bi-clustering · Gene expression data · Microarray technology · Multi-objective optimization · AMOSA · Point symmetry-based distance · Line symmetry-based distance

1 Introduction

Gene expression data analysis through microarray technology helps to solve different problems like medical diagnosis, biomedicine, gene expression profiling. A key step toward this analysis is the automatic identification of a group of genes that exhibit similar expression patterns. Clustering (Jain and Dubes 1988) is an effective way for finding out homogeneous groups of genes having similar expression profiles under different experimental conditions. Clustering techniques are popularly used in the field of microarray data analysis where genes are partitioned into different clusters depending on similarity measured based on the expression values. The primary assumption behind the clustering of microarray data is that genes belonging to a single cluster should have similar behaviors over all experimental conditions of that microarray data. But as the number of attributes or conditions increases, it is very unlikely to extract groups of genes possessing similar activities over all experimental conditions (Cheng and Church 2000). Clustering over all conditions may miss relationships present only over a locality of attributes. As a remedy, bi-clustering techniques are proposed in the literature to identify local patterns present in any gene expression dataset (Liu et al. 2008; Bryan et al. 2005).

A bi-cluster of a gene expression dataset is basically a subset of genes which show similar expression patterns over a subset of conditions (Hartigan 1972) (not over all conditions as done in conventional clustering).

Like any clustering algorithm, the effectiveness of bi-clustering algorithms also depends on the underlying dis-

Communicated by V. Loia.

First two authors have equal contributions.

✉ Sudipta Acharya
sudiptaacharya.2012@gmail.com

Sriparna Saha
sriparna.saha@gmail.com

Pracheta Sahoo
pracheta.sahoo@gmail.com

[1] Department of Computer Science and Engineering, Indian Institute of Technology, Patna, India

[2] Department of Mathematics and Computing, Indian Institute of Technology, Patna, India

tance measure for allocation of data points to different bi-clusters. Literature survey shows that commonly used distance measure in most of the clustering and bi-clustering algorithms is Euclidean distance (Acharya et al. 2016; Sahoo et al. 2016). But in recent years various symmetry-based distance measures are proposed for data clustering (Ray et al. 2007). Symmetry can be of various types like point symmetry, line symmetry, plane symmetry. In order to capture different types of symmetry present in clusters, different forms of symmetry-based distances are also proposed in the literature like point symmetry-based distance (Bandyopadhyay and Saha 2007), line symmetry-based distance Bandyopadhyay and Saha (2012). These distances are also proven to perform well for detecting clusters from gene expression datasets (Acharya et al. 2016; Acharya and Saha 2016). Motivated by this, in the current work, efforts are also made in developing some bi-clustering algorithms utilizing various symmetry-based distances. Moreover, as bi-clustering task requires to optimize several quality measures simultaneously, the problem is formulated as a multi-objective optimization or MOO problem. In this paper in order to make this simultaneous optimization feasible and also to identify a set of trade-off solutions, the search capability of a popular MOO technique, archived multi-objective simulated annealing (AMOSA) (Bandyopadhyay et al. 2008) is accomplished. It has been shown in the literature that AMOSA excels in the field of MOO as compared to several other existing multi-objective evolutionary algorithms.

Motivated by the above-mentioned facts in this paper we have proposed AMOSA-based bi-clustering algorithm utilizing three different distance measures, point symmetry (Bandyopadhyay and Saha 2007), line symmetry (Bandyopadhyay and Saha 2012) and traditional Euclidean distance, to mine coherent patterns from microarray data.

There are a number of existing works proposing different bi-clustering techniques. In Hartigan (1972), the concept of bi-clustering was first emerged as direct clustering. In Cheng and Church (2000) authors have shown the use of mean squared residue (MSR) for the calculation of coherent bi-clusters where the algorithm follows a heuristic greedy search technique. In this process, bi-clusters are identified in an iterative manner. For hierarchical clustering in both dimensions, a CTWC, i.e., a coupled two-way clustering method, is shown in Getz et al. (2000). Other well-known bi-clustering techniques in the existing state of the art are random walk-based bi-clustering (RWB) (Angiulli and Pizzuti 2005), simulated annealing-based bi-clustering (Bryan et al. 2005), order preserving submatrix algorithm (OPSM) (Ben-Dor et al. 2003), iterative signature algorithm (ISA) (Ihmels et al. 2004), BiVisu (Cheng et al. 2007), etc. A two-phase probabilistic bi-clustering model termed as FLOC was evolved by Yang et al. (2003) to simultaneously discover a set of possibly overlapping bi-clusters. On the other side, efficient techniques have been successfully amalgamated in the deterministic bi-clustering with frequent pattern mining algorithm (DBF) (Zhang et al. 2004).

In Chakraborty and Maka (2005) authors have proposed a single-objective genetic algorithm-based bi-clustering technique for gene expression datasets. In Maulik et al. (2009) a multi-objective optimization-based bi-clustering technique, MOGAB, was implemented in order to have solutions from gene expression datasets. Here NSGA-II (Deb et al. 2002) was used as the underlying optimization algorithm. In Chakraborty and Maka (2005) and Maulik et al. (2009) authors have adopted genetic algorithm-based optimization technique, which is a bio-inspired optimization technique, for bi-clustering the gene expression dataset. In Liu et al. (2008) authors have proposed a multi-objective particle swarm optimization-based bi-clustering (MOPSOB) algorithm to find out coherent bi-clusters from gene expression data. In Seridi et al. (2015) authors have proposed a new multi-objective model which aims to extract bi-clusters with maximal size and biological relevance. They proposed a multi-objective hybrid metaheuristic HMOBI based on multi-objective evolutionary algorithm, IBEA, and a dominance-based multi-objective local search, DMLS. HMOBI takes advantages of the evolutionary and local search metaheuristic behaviors. Another evolutionary algorithm SMOB was proposed by Divina and Aguilar-Ruiz (2007) where a weighted sum of three functions: MSR, volume and row variance (RV), is used as the objective function. Recently, in Bousselmi et al. (2017) authors have developed a new multi-objective bi-clustering algorithm, called Bi-MOCK, which is an extended version of the multi-objective clustering algorithm MOCK. Bi-MOCK inherits the main feature of MOCK, which is the ability to determine the number of bi-clusters automatically. Another evolutionary algorithm (EA)-based bi-clustering algorithm was proposed by Huang et al. (2012). Here authors have proposed a new bi-clustering algorithm based on the use of an EA together with hierarchical clustering. The authors argued that with such a huge search space, the EA itself will not be able to find optimal or approximately optimal solutions within a reasonable time. Therefore, they proposed to separate the conditions into a number of condition subsets, also called subspaces. The evolutionary algorithm is then applied to each subspace in parallel, and an expanding and merging phase is finally employed to combine the results of subspaces into the output bi-clusters. As it is related only to the conditions, the EA is called condition-based evolutionary bi-clustering (CBEB), where the normalized geometric selection method is used as the selection function and the simple crossover and binary mutation methods are employed for reproducing the offspring. Apart from evolutionary algorithms, in Hochreiter et al. (2010) authors have introduced a novel bi-clustering method that is a generative multiplicative model.

It assumes realistic non-Gaussian signal distributions with heavy tails. The generative model allows to rank bi-clusters according to their information content. Another bi-clustering technique based on related genes and condition extraction was proposed by Yan and Wang (2013). In order to reduce the computational complexity and interference between different types of bi-clusters, related genes and conditions are extracted for each type of bi-clusters by constructing sparse matrices. Hierarchical clustering algorithm is employed to obtain candidate bi-clusters from the sparse matrices. Concept of multi-objective optimization was not used in Yan and Wang (2013). After conducting thorough literature survey the following observations were made: (1) The efficacy of symmetry-based distances was never explored in the field of bi-clustering. But in recent years symmetry-based distances are found to perform well for performing the clustering task. Thus, application of symmetry-based distances in solving the bi-clustering task can be a logical consequence. (2) Most of the existing multi-objective bi-clustering approaches utilize search capability of evolutionary computation for optimizing multiple bi-cluster quality measures simultaneously. Recent study (Acharya et al. 2016; Acharya and Saha 2016) revealed that simulated annealing-based multi-objective technique, AMOSA, performs better than existing multi-objective evolutionary techniques, namely NSGA-II, PAES, etc. But utility of AMOSA in the bi-clustering task was never explored. These facts motivate us to utilize symmetry-based distances in developing a multi-objective-based bi-clustering technique whereas the underlying optimization strategy, archived multi-objective simulated annealing (AMOSA) (Bandyopadhyay et al. 2008), is used. In the domain of optimization theory, AMOSA (Bandyopadhyay et al. 2008) is indeed a promising technique. It is a proven fact that NSGA-II (Deb et al. 2002) and some other MOO-based problems underperform as compared to AMOSA (Bandyopadhyay et al. 2008) in solving different benchmark test problems. Inspired by these facts, in the current research work, a bi-clustering technique is proposed which uses the search capacity of AMOSA. Our proposed bi-clustering technique simultaneously performs clustering on both genes and conditions of three gene microarray datasets. Also, we have alternatively used three different distance measures as the underlying proximity measures in our proposed algorithm. The motivations of using these three distance functions are as follows.

- As supported by the existing literature, Euclidean distance has been used widely in performing clustering on biological datasets (Acharya et al. 2016). Therefore, in this work, one of our chosen distance measures is Euclidean distance.
- To recognize or identify real-life objects, the property of 'symmetry' is very useful (Attneave 1955). Similar to the real-life objects, symmetricity can also be seen in the structures of clusters/bi-clusters. Symmetry measurement can be of two types, point symmetry (PS) (Bandyopadhyay and Saha 2012) and line symmetry (LS) (Bandyopadhyay and Saha 2012).
- The symmetricity of clusters/bi-clusters with respect to a central point can be measured with the help of point symmetry-based distance (Bandyopadhyay and Saha 2012). Literature survey supports the use of this distance function in performing clustering on gene expression datasets (Acharya et al. 2016; Acharya and Saha 2016).
- But if a cluster/bi-cluster is not symmetric about a central point but symmetric about a median line (called principal component Bandyopadhyay and Saha 2012) of that cluster, then point symmetry-based measurements cannot capture the symmetricity of that cluster. To detect these types of clusters, line symmetry (Bandyopadhyay and Saha 2012)-based measurements are most effective.
- In Acharya and Saha (2016), it was experimentally shown that point/line symmetry-based distances perform well in automatically clustering gene expression datasets and mi-RNA datasets compared to some traditional distance functions. This also motivates us to use symmetry-based distance in the bi-clustering process.

The major contributions of the current work are enumerated below.

- Use of symmetry (line or point)-based distance measure in the bi-clustering process was never explored in the existing literature. Point and line symmetry-based distances (Bandyopadhyay and Saha 2012) are capable of identifying symmetric clusters/bi-clusters, which is not possible with the help of traditional distance measures like Euclidean. Therefore, symmetry-based measures possess some extended capabilities in identifying clusters/bi-clusters. In this paper we have performed a comparative study by solving the bi-clustering task utilizing three distance measures namely Euclidean, point symmetry-based distance (Bandyopadhyay and Saha 2012) and line symmetry-based distance (Bandyopadhyay and Saha 2012). Our aim is to show how symmetry-based distances outperform traditional distance measures like Euclidean distance which is used by most of the existing bi-clustering algorithms.
- Another contribution of the paper is to show the utility of popular multi-objective optimization technique, AMOSA (Bandyopadhyay et al. 2008), in solving the bi-clustering problem. Two widely used bi-cluster quality measures, mean squared residue (MSR) (Maulik et al. 2009) and row variance (RV) (Maulik et al. 2009), are optimized simultaneously using the search capability of AMOSA.

– Using these three different distance measures as mentioned above, three different versions of the proposed bi-clustering algorithm named as *Eucli-AMOSAB* (AMOSA-based bi-clustering using Euclidean distance), *PS-AMOSAB* (AMOSA-based bi-clustering using point symmetry-based distance), *LS-AMOSAB* (AMOSA-based bi-clustering using line symmetry-based distance) are developed.

2 Formulation of bi-clustering problem for gene expression data

Multi-objective optimization (MOO) problems require more than one objective functions to be optimized simultaneously. The details of MOO technique can be found at Deb et al. (2016). As our proposed bi-clustering technique is multi-objective in nature, the first task is to select some good bi-cluster quality measures which can be simultaneously optimized by any available MOO technique. This section deals with the formulation of this problem, i.e., bi-clustering of gene expression data as a MOO problem. We have used AMOSA (Bandyopadhyay et al. 2008) as the underlying optimization strategy for our proposed bi-clustering technique. Within AMOSA we have utilized two widely used bi-cluster quality measures as two objective functions, mean squared residue (Cheng and Church 2000) and row variance (Cheng and Church 2000). These measures are developed by Cheng and Church (Cheng and Church 2000). If a gene expression dataset is represented by matrix G with T number of genes and V number of conditions, then a bi-cluster is represented as a submatrix $B(P, Q)$ of matrix G, where $P \subseteq T$ and $Q \subseteq V$. Each element g_{ij} of matrix G corresponds to the expression level of the ith gene at the jth condition.

Volume of bi-cluster is calculated as follows.
volume $= vol(P, Q) = |P| \times |Q|$.

The chosen objective functions as mentioned above are described below.

2.1 Mean squared residue

The mean squared residue introduced by Cheng and Church (2000) has become one of the most popular measures to identify bi-clusters in most of the bi-clustering algorithms. We define the MSR of a bi-cluster $B(P, Q)$ as follows.

$$\mathrm{MSR}(P, Q) = \frac{1}{\mathrm{vol}(P, Q)} \sum_{i \in P, j \in Q} (a_{ij} - a_{iQ} - a_{Pj} + a_{PQ})^2 \quad (1)$$

where $a_{iQ} = (1/|Q|) \sum_{j \in Q} a_{ij}$, $a_{Pj} = (1/|P|) \sum_{i \in P} a_{ij}$, and $a_{PQ} = (1/(|P| \times |Q|)) \sum_{i \in P, j \in Q} a_{ij}$, i.e., a_{iQ}, a_{Pj} and a_{PQ} represent the mean value of the ith row, the mean value of the jth column and the mean value of the elements in the given bi-cluster, respectively. According to Cheng and Church (2000) the residue of an element a_{ij} in a submatrix B equals,

$$r_{ij} = a_{ij} - a_{iQ} - a_{Pj} + a_{PQ}$$

The difference between the actual value of a_{ij} and its expected value predicted from its row, column and bi-cluster mean is given by the residue of an element. It also reveals its degree of coherence with the other entries of the bi-cluster. The quality of a bi-cluster can be evaluated by computing the mean squared residue $\mathrm{MSR}(P, Q)$, i.e., the sum of all the squared residues of its elements (Cheng and Church 2000).

Lower the score means larger is the coherence, and better is the richness of the bi-cluster. A bi-cluster $B(P, Q)$ is called a δ bi-cluster if $\mathrm{MSR}(P, Q) < \delta$, a known threshold value.

2.2 Row variance

The definition of row variance $\mathrm{RV}(P, Q)$ of a bi-cluster $B(P, Q)$ is,

$$\mathrm{RV}(P, Q) = \frac{1}{\mathrm{vol}(P, Q)} \sum_{i \in P, j \in Q} (a_{ij} - a_{iQ})^2 \quad (2)$$

Bi-clusters with high row variance are more interesting because those make significant changes in the expression levels of the genes. The aim is to search for the bi-clusters having high average row variance and MSR score below a threshold level δ.

The calculation of MSR and RV can be illustrated with a simple example. Suppose a bi-cluster is represented as follows.

$$B = \begin{bmatrix} 1 & 1 & 2 \\ 3 & 3 & 3 \\ 2 & 1 & 2 \\ 2 & 3 & 1 \end{bmatrix}$$

where $P = 4$ (number of rows) and $Q = 3$ (number of columns). a_{iQ}, by varying i, i.e., mean of each row, is calculated as follows.

$a_{13} = 1.33, a_{23} = 3, a_{33} = 1.66$ and $a_{43} = 2$.

Similarly, a_{Pj} by varying j, i.e., mean of each column, is calculated and values are as follows.

$a_{41} = 2, a_{42} = 2$ and $a_{43} = 2$.

a_{PQ}, i.e., mean of whole bi-cluster is 2.

$$\text{vol}(P, Q) = (4 \times 3) = 12.$$

Next for each element of the above bi-cluster the term $\sum_{i\in P, j\in Q}(a_{ij} - a_{iQ} - a_{Pj} + a_{PQ})^2$ is calculated and its value is 3.3335. So dividing this term by vol(P, Q) (according to Eq. 1) we find MSR(P, Q), i.e., MSR(4, 3) $= \frac{3.3335}{12}$=0.277.

Similarly, to calculate RV(P, Q), i.e., RV(4, 3), we calculate the term $\sum_{i\in P, j\in Q}(a_{ij} - a_{iQ})^2$ and its value is 3.3335. So, the obtained RV measure, RV(4, 3) = 0.277.

For the above example of bi-cluster, the obtained MSR and RV values are the same. But this scenario is not very common.

2.3 AMOSA: underlying optimization strategy

In our proposed bi-clustering algorithm, AMOSA (Bandyopadhyay et al. 2008), which is a generalized version of probabilistic metaheuristic-based simulated annealing (SA), is utilized as the underlying optimization technique. Simulated annealing is a popular single-objective optimization technique based on the properties of statistical mechanics. There are very few works where SA is extended to solve the multi-objective optimization problems (Bandyopadhyay et al. 2008). It is because SA follows the strategy of search from a point and produces only a single solution after one run. Moreover, it is also difficult to compute the energy difference between two solutions required for SA in case of multiple objective functions. To overcome this limitation, a multi-objective-based approach is developed based on the search properties of SA, namely archived multi-objective simulated annealing (AMOSA) (Bandyopadhyay et al. 2008). Many new concepts are incorporated in AMOSA. An archive is maintained where all the non-dominated solutions found during the search process are stored. Here two limits are kept on the size of the archive, soft limit (SL) and hard limit (HL). Generally SL $>$ HL. During the search process, non-dominated solutions are stored in the archive up to the limit of SL. Once the number of solutions crosses SL, a single linkage-based clustering is applied on it to reduce the size to HL.

In AMOSA, we kept initial temperature as $T_{\max}$. Initially as the solutions of the archive are generated randomly. A single solution is picked randomly from the archive, and it is called the *c-pt*. Now, to generate a new solution, *n-pt*, mutation is performed on the *c-pt*. Thereafter, objective function values are calculated for these two solutions and domination status between *c-pt* and *n-pt* is checked. In order to determine the amount of domination between two solutions, a new formula is proposed in Bandyopadhyay et al. (2008). The amount of domination $\Delta\text{dom}(a, b)$, between two solutions a and b, is defined as follows.

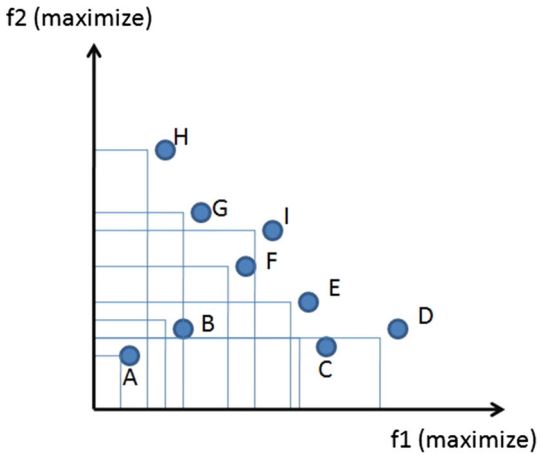

Fig. 1 Pareto-optimal front and different domination examples

$$\Delta\text{dom}_{a,b} = \prod_{i=1, f_i(a)\neq f_i(b)}^{M_{\text{obj}}} \frac{|f_i(a) - f_i(b)|}{R_i} \tag{3}$$

where $f_i(a)$ and $f_i(b)$ are the ith objective values of the two solutions. The range of the ith objective is denoted by R_i, and M_{obj} denotes the number of objective functions. Now based on the domination status of the *n-pt* and the *c-pt*, three different cases can arise.

Case 1 In this case, *n-pt* is dominated or non-dominated by *c-pt*, and it may be dominated by some points present in the archive. Let total number of dominated points excluding the *c-pt* in the archive be k. Such cases are illustrated in Fig. 1. Points D, E, F, G and H are some active points and available at any instant in the archive. Let *c-pt* be F, and the *n-pt* be B. Then a quantity $\Delta\text{dom}_{\text{avg}}$ is computed as follows.

$$\Delta\text{dom}_{\text{avg}} = \frac{(\sum_{i=1}^{k}(\Delta\text{dom}_{i,n-pt}) + \Delta\text{dom}_{c-pt,n-pt})}{(k+1)} \tag{4}$$

The *n-pt* can be accepted as the *c-pt* with the following probability

$$\text{probability} = \frac{1}{1 + e^{\Delta\text{dom}_{\text{avg}} \times \text{tmp}}} \tag{5}$$

Here, average amount of domination between *n-pt* and $(k+1)$ points is denoted by $\Delta\text{dom}_{\text{avg}}$. These $(k+1)$ points include *c-pt* and some k points in the archive.

Case 2 In this case, the *n-pt* is non-dominating with respect to *c-pt* and some points in the archive. Here in Fig. 1, this case is shown. For example, let *c-pt* be denoted by F and E be the *n-pt*, *c-pt* be G and I be the *n-pt*. If any point in the archive is dominated by the *n-pt*, then it is removed from the archive. If *n-pt* is non-dominated with respect to all or few points

in the archive, then accept it as *c-pt* and add it to the final archive. After the addition of this point if archive overflows, then single linkage clustering technique (Seifoddini 1989) is applied to reduce the archive size to HL.

Case 3 In this case *c-pt* is dominated by the *n-pt* and *n-pt* is dominated by some k points in the archive. Now suppose in Fig. 1, *c-pt* be A and *n-pt* be B. The minimum of difference of amount of domination between *n-pt* and some k points in the archive is denoted by $\Delta\text{dom}_{\min}$. The point which corresponds to the minimum difference from the archive is accepted as the *c-pt* with the probability,

$$\text{probability} = \frac{1}{1 + \exp(-\Delta\text{dom}_{\min})} \tag{6}$$

Otherwise the *n-pt* is selected as the *c-pt*. The above-mentioned process is repeated for *TotalIter* times for each temperature (tmp). With the help of cooling rate α, temperature is reduced as $\alpha \times \text{tmp}$, till the minimum temperature $T_{\min}$ is attained. Thereafter, the process stops and finally archive contains the final non-dominated solutions. In Bandyopadhyay et al. (2008), it has been shown that the performance of AMOSA is better than NSGA-II (Deb et al. 2002) and some other multi-objective-based optimization techniques. Inspired by this observation in the current work we have used AMOSA as the underline optimization technique. The pseudo-code of the AMOSA algorithm is shown in Fig. 2.

2.4 Chosen distance measures

For any clustering/bi-clustering algorithm, its efficacy majorly depends on the underlying distance measure used by that algorithm to allocate data points to different clusters/bi-clusters. From different existing literature (Dudoit and Fridlyand 2003; Giancarlo et al. 2010; Ray et al. 2007) it has been very clear that choosing proper distance measure is a challenging task. Inspired by this fact, we have decided to perform a comparative study between three different distance measures while performing the bi-clustering task. We have chosen popular Euclidean, point symmetry (PS) (Bandyopadhyay and Saha 2012) and line symmetry (Bandyopadhyay and Saha 2012)-based distance as proximity measures. The mathematical formulations of chosen distance measures are given as follows.

2.5 Euclidean distance

Euclidean distance between two points x_i and x_j is defined as,

$$d_e(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \tag{7}$$

where d is the dimension of each data point.

2.6 Point symmetry-based distance

The point symmetry-based distance or PS distance (Bandyopadhyay and Saha 2012) between a point x_i of lth cluster and its centroid c_l is represented by $d_{\text{ps}}(x_i, c_l)$ which is defined as follows.

To compute PS distance, we need to find the reflected point of x_i with respect to the cluster center c_l. Let the reflected point be denoted by $x_i{}^*$. Then $x_i{}^* = (2 * c_l - x_i)$. The PS distance between x_i and the center c_l is defined as,

$$d_{\text{ps}}(x_i, c_l) = d_{\text{sym}}(x_i, c_l) \times d_e(x_i, c_l) \tag{8}$$

where

$$d_{\text{sym}}(x_i, c_l) = \frac{\sum_{i=j}^{\text{knear}} d_j}{\text{knear}} \tag{9}$$

where knear unique nearest neighbors of x_i^* are at Euclidean distances of d_j, $j = 1, 2, \ldots$ knear, respectively. $d_e(x_i, c_l)$ is the Euclidean distance between the point x_i and the cluster center, c_l of the lth cluster. According to Eq. 9, knear should not be equal to 1 because if x_i exists in the dataset, then $d_{\text{ps}}(x_i, c_l) = 0$. In that case there will be no impact of the Euclidean distance on the point symmetry distance computation. Also, large values of knear may not be suitable because it may overestimate the amount of symmetry of a point with respect to a particular cluster center. In Bandyopadhyay and Saha (2012) and Bandyopadhyay and Saha (2007), authors have experimentally proved that with knear $= 2$, point symmetry-based distance performs the best in detecting different shaped clusters. In Bandyopadhyay and Saha (2012) and Bandyopadhyay and Saha (2007), the value of knear was varied over a range and results were taken for different values. Their experimentation showed that with knear $= 2$ obtained clustering results were the best.

The notion of point symmetry-based distance can be understood with a figure as shown in Fig. 3. Suppose there is a point $\overline{q}$ in a cluster as shown in Fig. 3. The center of the cluster is denoted by $\overline{cn}$. As it can be seen from figure, the reflected point of $\overline{q}$ with respect to $\overline{cn}$ is $\overline{q^*}$ which is calculated as $\overline{q^*} = 2 \times \overline{cn} - \overline{q}$. Suppose two nearest neighbors of $\overline{q^*}$ are at Euclidean distances of d_1 and d_2, respectively. The point symmetry-based distance between $\bar{q}$ and $\bar{cn}$ is therefore calculated as $d_{\text{ps}}(\bar{q}, \bar{cn}) = \frac{d_1+d_2}{2} \times d_e(\bar{q}, \bar{cn})$.

2.7 Line symmetry-based distance

In line symmetry-based distance (Bandyopadhyay and Saha 2012) to measure the amount of line symmetry (LS) of a point (x_i) of lth cluster with respect to its symmetric axis or

Set T_{max}, T_{min}, *HL*, *SL*, *TotalIter*, α, tmp=T_{max}.
Initialization of *Archive*.
c-pt = random(*Archive*). /* solution chosen randomly from *Archive**/
while (tmp $> T_{min}$)
  for (i=0; i< TotalIter; i++)
    n-pt=perturb(*c-pt*).
    Checking domination status of *n-pt* and *c-pt*.
    /* Code for three different cases */
    if (*c-pt* dominates *n-pt*) /* Case 1*/
      $\Delta dom_{avg} = \frac{\left(\sum_{i=1}^{k}(\Delta dom_{i,n-pt}) + \Delta dom_{c-pt,n-pt}\right)}{(k+1)}$.
      /* *k*= total number of points in the *Archive* which dominate *n-pt*, $k \geq 0$. */
      Set *n-pt* as *c-pt* with probability as shown in Equation (5).
    if (*n-pt* and *c-pt* are non-dominating to each other) /* Case 2*/
      Check the domination status of *n-pt* and points in the *Archive*.
      if (*n-pt* is dominated by k ($k \geq 1$) points in the *Archive*) /* Case 2(a)*/
        $\Delta dom_{avg} = \frac{\left(\sum_{i=1}^{k} \Delta dom_{i,n-pt}\right)}{k}$.
        Set *n-pt* as *c-pt* with probability as shown in Equation (5).
      if (*n-pt* is non-dominating w.r.t all the points in the *Archive*) /* Case 2(b)*/
        Set *n-pt* as *c-pt* and add *n-pt* to the *Archive*.
        if *Archive-size* $>$ *SL*
          Cluster *Archive* to *HL* number of clusters.
      if (*n-pt* dominates *k*, ($k \geq 1$) points of the *Archive*) /* Case 2(c)*/
        Set *n-pt* as *c-pt* and add it to *Archive*.
        Remove all the *k* dominated points from the *Archive*.
    if (*n-pt* dominates *c-pt*) /* Case 3 */
      Check the domination status of *n-pt* and points in the *Archive*.
      if (*n-pt* is dominated by k ($k \geq 1$) points in the *Archive*) /* Case 3(a)*/
        Δdom_{min} = minimum of the difference of domination amounts between the *n-pt* and the k points
        prob= $\frac{1}{1+exp(-\Delta dom_{min})}$.
        Set point of the archive which corresponds to Δdom_{min} as
        c-pt with probability=*prob*.
        else set *n-pt* as *c-pt*
      if (*n-pt* is non-dominating with respect to the points in the *Archive*) /* Case 3(b) */
        select the *n-pt* as the *c-pt* and add it to the *Archive*.
        if *c-pt* is in the *Archive*, remove it from *Archive*.
        else if *Archive-size*> *SL*.
          Cluster *Archive* to *HL* number of clusters.
      if (*n-pt* dominates *k* other points in the *Archive*) /* Case 3(c)*/
        Set *n-pt* as *c-pt* and add it to the *Archive*.
        Remove all the k dominated points from the *Archive*.
  End for
$tmp = \alpha * tmp$.
End while
if *Archive-size* $>$ *SL*
  Cluster *Archive* to *HL* number of clusters.

Fig. 2 Pseudo-code of AMOSA algorithm

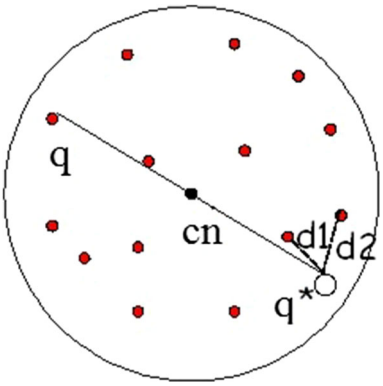

Fig. 3 Example of point symmetry-based distance

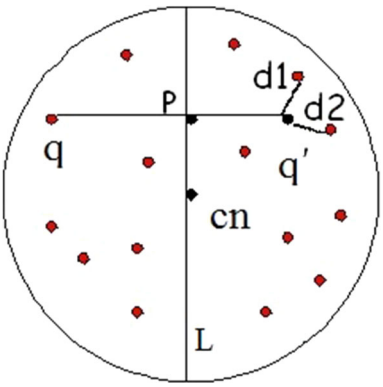

Fig. 4 Example of line symmetry-based distance

first principal component, L, denoted as, $d_{\text{ls}}(x_i, L)$, the steps which should be followed are given below.

1. For a data point x_i, its projected point p_i on the symmetric axis L is calculated.
2. Find $d_{\text{sym}}(x_i, p_i)$ as:

$$d_{\text{sym}}(x_i, p_i) = \frac{\sum_{j=1}^{\text{knear}} d_j}{\text{knear}} \tag{10}$$

where knear unique nearest neighbors of reflected point $x_i^* = 2 \times p_i - x_i$ are at Euclidean distances of d_j, $j = 1, 2, \ldots,$ knear. Then $d_{\text{ls}}(x_i, L)$ is calculated below.

$$d_{\text{ls}}(x_i, L) = d_{\text{sym}}(x_i, p_i) \times d_e(x_i, c_l) \tag{11}$$

According to Bandyopadhyay and Saha (2012, 2007) here also we have chosen knear $= 2$.

The above-mentioned line symmetry-based distance can be understood more clearly from Fig. 4. In the figure, let $\bar{q}$ be a point within a cluster and $\bar{cn}$ be its center point. L represents the symmetric axis of lth cluster. Let the projected point of $\bar{q}$ on this symmetric axis L be denoted by $\bar{p}$, and the reflected point of $\bar{q}$ with respect to $\bar{p}$ be $\bar{q'}$. The first two nearest neighbors (here knear is chosen equal to 2) of $\bar{q'}$ are at Euclidean distances of d_1 and d_2, respectively. Therefore, the total amount of line symmetry of $\bar{q}$ with respect to the projected point $\bar{p}$ on L is calculated as: $d_{\text{sym}}(\bar{q}, \bar{p}) = \frac{d_1+d_2}{2}$. Therefore, the total line symmetry-based distance of point $\bar{q}$ with respect to the symmetric axis of lth cluster is calculated as $d_{\text{ls}}(\bar{q}, L) = d_{\text{sym}}(\bar{q}, \bar{p}) \times d_e(\bar{q}, \bar{cn})$, where $d_e(\bar{q}, \bar{cn})$ is the Euclidean distance between the point $\bar{q}$ and the cluster center $\bar{cn}$. Our proposed bi-clustering technique is generic in nature, i.e., any other distance measure can be used to allocate data points to different bi-clusters. In this work we have chosen three above-mentioned distance measures and performed a comparative study between the three versions while incorporated in our proposed bi-clustering algorithm.

3 Proposed archived multi-objective simulated annealing-based bi-clustering technique

Based on the optimization strategy of AMOSA (Bandyopadhyay et al. 2008), different steps of our proposed bi-clustering technique are given below. The flowchart of our proposed bi-clustering algorithm is also given in Fig. 7.

3.1 String or solution representation and archive initialization

The proposed bi-clustering approach starts with the method of initialization where the archive (which is a term associated with AMOSA) is initialized by some randomly chosen solutions. Here each solution or archive_element is encoded as a string of centroids.

As it is a bi-clustering technique, so we represent each archive_element by two components: One is dedicated for clustering the genes, and another for the clustering of conditions.

Now for a $T \times V$ microarray dataset, the number of genes is T and the number of conditions is V.

If no domain information is available, a dataset containing total n points can have at most $\sqrt{(n)}$ clusters. Similarly, for a $T \times V$ microarray dataset, $|\sqrt{T}|$ and $|\sqrt{V}|$, respectively, represent the maximum number of gene clusters (M) and condition clusters (N). Therefore, the range of M lies between 2 and $|\sqrt{T}|$, and the range of N lies between 2 and $|\sqrt{V}|$, which in turn makes the length of the string ranging from 4 to $(|\sqrt{T}| + |\sqrt{V}|)$.

Each gene or condition has the same probability to become a gene or condition cluster representative during initialization.

The number of gene cluster centroids, M, encoded in a string i is selected randomly in the range of two extreme

points, i.e., $K_{\min}$=2 and $K_{\max 1} = |\sqrt{(T)}|$. Similarly the number of condition cluster centroids, N, in ith string is selected randomly between $K_{\min} = 2$ and $K_{\max 2} = |\sqrt{(V)}|$. The following equations explain how the random values are generated:

$$M = (\text{rand}()\text{mod}(K_{\max 1} - 1)) + 2$$
$$N = (\text{rand}()\text{mod}(K_{\max 2} - 1)) + 2$$

Here, rand() is a function which generates a random integer number and $K_{\max 1}$, $K_{\max 2}$ are the upper limits of the number of genes and condition clusters, respectively. The standard rand() function of C language (available at stdlib.h header file) is used here. rand() returns a pseudo-random integral number in the range between 0 and RAND_MAX (constant defined in header file stdlib.h, its value depends on library used in the compiler, but in general it is guaranteed to be at least 32767). The pseudo-random number is generated by an algorithm that returns a sequence of apparently non-related numbers each time it is called.[1] This algorithm uses a seed to generate the series, which should be initialized to some distinctive value using function srand. rand() function works very similar to a uniform distribution[2] in the range 0 to RAND_MAX.

Mathematically, the range of M and N can be written as $(2 \leq M \leq K_{\max 1})$ and $(2 \leq N \leq K_{\max 2})$, respectively. In Fig. 5 the encoded strings are shown.

3.2 Assignment of data points and updation of cluster centers

The archive is initialized with randomly selected gene and condition cluster centers, followed by the assignment of T number of genes to different gene clusters and assignment of V number of conditions to different condition clusters.

Each gene or condition is placed to corresponding minimum distant gene or condition cluster according to the following formulas.

$$k = \text{argmin}_{k=1\ldots M} d(g_i, gt_k)$$
$$l = \text{argmin}_{l=1\ldots N} d(c_j, ct_l)$$

Here, M and N represent the number of gene clusters and the number of condition clusters, respectively, encoded in one archive_member.

gt_k and ct_l denote the center of kth gene cluster and lth condition cluster, respectively.

g_i or c_j is a particular gene or condition, respectively.

$d(g_i, gt_k)$ denotes distance between kth gene cluster center and ith gene, g_i.

$d(c_j, ct_l)$ denotes distance between lth condition cluster center and jth condition, c_j.

In three different versions of our proposed bi-clustering algorithm, the above two distance functions are calculated according to the adopted distance measure (Euclidean, PS or LS according to Eqs. 7, 8 and 11, respectively).

k and l are the index of gene cluster or condition cluster, respectively, in which the gene g_i or condition c_j should be assigned, i.e., index of that gene/condition cluster with respect to which the gene g_i or condition c_j has minimum distance.

After allocation of data points to different clusters, updation of cluster centers is performed. The most centrally located point from which all other points in the same cluster have the minimum average distance is selected as the new cluster center, i.e., either gene or condition cluster center is updated by the newly chosen point. In the upper example, suppose in ith gene cluster having center gt_i, first q number of genes of gene expression dataset are assigned. Now the center gt_i will be updated as follows.

$$\forall_{k=1}^{V} gt_i[k](\text{updated}) = \frac{\sum_{j=1}^{q}(\text{arr}[g_j][c_k])}{q},$$

where V is the dimension or number of conditions for each gene and $gt_i[k]$(updated) is the kth dimension of updated ith gene cluster. g_j represents jth gene, and c_k represents the kth condition. $arr[g_j][c_k]$ denotes the expression value of gene expression dataset for jth gene and kth condition.

3.3 Generating bi-clusters from archive_element

For a valid string or archive_element with no recurrence of gene and condition cluster indices, and $2 \leq M \leq \lceil\sqrt{T}\rceil$ and $2 \leq N \leq \lceil\sqrt{V}\rceil$), at first we need to extract all the gene clusters and the condition clusters encoded in that string. For example, consider the second archive_element or solution shown in Fig. 5. The decoded bi-clusters from this solution are as follows.

$< gt_1\ ct_1 >, < gt_1\ ct_2 >, < gt_1\ ct_3 >, < gt_1\ ct_4 >, < gt_2\ ct_1 >, < gt_2\ ct_2 >, < gt_2\ ct_3 >, < gt_2\ ct_4 >$. Here each bi-cluster is represented by the pair of gene cluster center and condition cluster center.

3.4 Calculation of objective functions

For the evaluation of bi-clusters, two objective functions, namely MSR and RV, are considered. Our target is to

[1] (http://www.cplusplus.com/reference/cstdlib/rand/)

[2] http://promodel.com/onlinehelp/promodel/80/C-14%20-%20Rand().htm

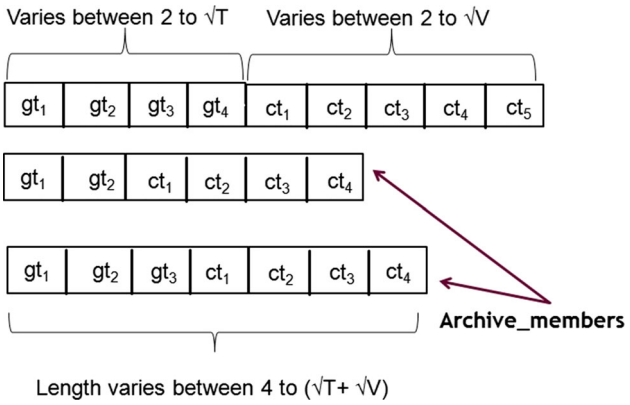

Fig. 5 String representation of proposed bi-clustering technique

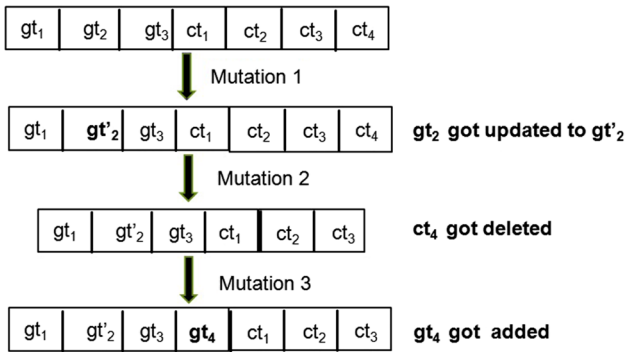

Fig. 6 Three types of mutation operations applied on each bi-clustering solution

minimize MSR and maximize RV to attain good-quality bi-clusters. The objective functions of a bi-cluster $B(P, Q)$ having P number of genes and Q number of conditions are denoted as follows.

$$(\text{fv})_1(P, Q) = \frac{\text{MSR}(P, Q)}{\delta} \text{ and } (\text{fv})_2(P, Q) = \frac{1}{1 + \text{RV}(P, Q)}$$

where $(\text{fv})_1$ and $(\text{fv})_2$ denote the two fitness vectors, respectively.

In the denominator of $(\text{fv})_2$, the term 1 is added to avoid the generation of undefined terms while dividing the numerator by zero, when the remaining RV term = 0. To obtain highly coherent, functionally rich bi-clusters both the objective functions should be minimized. A bi-cluster $B(P, Q)$ is called a δ bi-cluster if $\text{MSR}(P, Q) < \delta$, a known threshold value. For each solution/archive_element after decoding all possible bi-clusters (according to Sect. 3.3) all possible δ-bi-clusters expressed as (gene cluster, condition cluster) pair are selected. For each encoded δ-bi-cluster, the fitness vector $f = \{(\text{fv})_1, (\text{fv})_2\}$ is calculated. The mean of the fitness vectors of all encoded δ-bi-clusters existing in that particular string represents the fitness vector of the mentioned string. The generation of invalid strings may be explained as the factor of randomness incurred while choosing the mutation operators. The invalid strings are given objective function vector $f = \{X, X\}$, where X is an arbitrary large number, and after successive executions of the algorithm, the invalid strings are discarded automatically. From each non-dominated string of the final archive all the δ-bi-clusters are taken out to form the final bi-clusters.

3.5 Search operators

In order to explore the search space, mutation operations are used in AMOSA to generate new solutions from the current solution. Here we have used three different mutation operators. These are defined below.

- Mutation 1: In this variant of mutation, the gene or the condition center is perturbed by a little amount. It should be kept in mind that both the gene and condition centers cannot be varied simultaneously; instead, any of them should be targeted at a time. As a result each cluster center is updated by a random variable derived from Laplacian distribution, $p(\epsilon) \propto e^{-\frac{|\epsilon-\mu|}{\delta}}$. The amount of mutation is quantified by the following scaling factors μ and λ. The value of λ is generally 1.0. Mutation is applied for all dimensions of a chosen center.
- Mutation 2: The main motivation for using this variant of mutation is to decrease the length of the string. As above, this mutation type also deals with only one center that is either a gene center or a condition center at a time. The selected center is deleted in order to reduce the number of gene or condition clusters represented in the string by 1.
- Mutation 3: This mutation works in just the opposite way than mutation 2. It is used to increase the number of gene or condition cluster centers. At a time either a gene cluster center or a condition cluster center is appended so as to increase the number of gene or condition cluster centers by 1.

Each string goes through any one of the above-mentioned types of mutation operations and generates a new string. For each string probability of going through any one of the above-mentioned mutation is the same, i.e., 0.33.

In Fig. 6 three types of mutations are shown.

3.6 Selecting best bi-clustering solution from final Pareto-optimal front

Any multi-objective optimization technique produces a set of non-dominated solutions on the final Pareto-optimal front

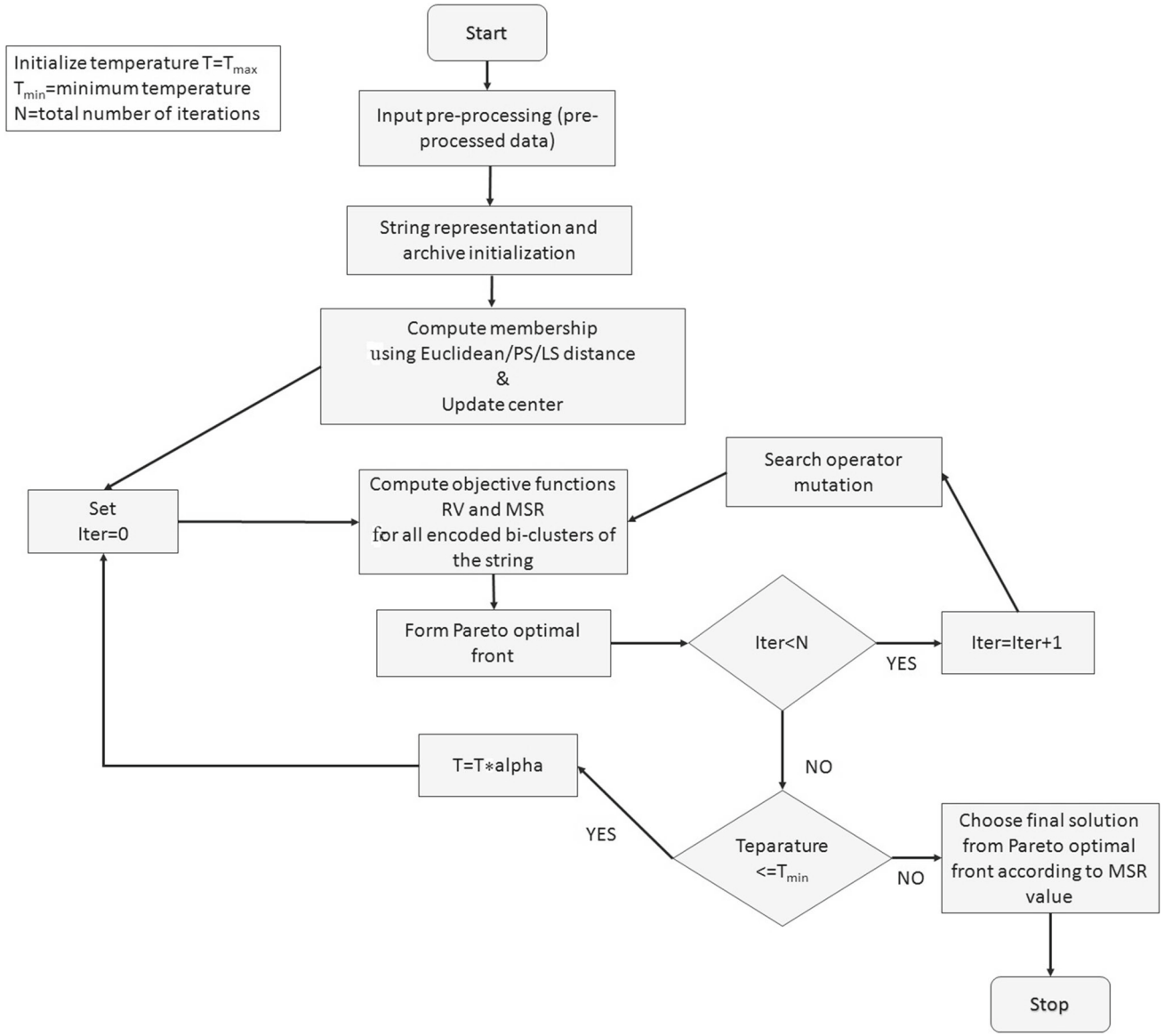

Fig. 7 Flowchart of proposed bi-clustering algorithm

(Deb et al. 2016). The best solution is chosen according to some criteria. Here the goodness of the solution is measured by its MSR (Maulik et al. 2009) value. The bi-clustering solution having the lowest average MSR value is chosen as the optimal solution.

3.7 Illustration with an example

Different steps of the proposed bi-clustering technique are illustrated with the following example:

Suppose a microarray data matrix G is provided as input as follows.

$$G = \begin{bmatrix} & c1 & c2 & c3 & c4 \\ g1 & 1 & 2 & 1 & 3 \\ g2 & 2 & 1 & 3 & 4 \\ g3 & 1 & 3 & 4 & 1 \\ g4 & 1 & 2 & 2 & 3 \\ g5 & 2 & 1 & 3 & 4 \\ g6 & 5 & 1 & 3 & 1 \\ g7 & 1 & 3 & 1 & 1 \\ g8 & 2 & 3 & 2 & 2 \\ g9 & 3 & 1 & 3 & 2 \end{bmatrix}$$

The data matrix is of size $T \times V$, where number of genes or $T = 9$ and number of conditions or $V = 4$. According to the first step, the archive is initialized with some random solutions. As $T = 9$ and $V = 4$, so according to Sect. 3.1, $2 \leq M \leq 3$ and $2 \leq N \leq 2$. Therefore, size of a bi-cluster will vary between 4 and $(3 + 2)$, i.e., 5. The steps of the proposed bi-clustering technique for the above input matrix are provided below.

Step 1 String representation and archive initialization: After randomly choosing values of M and N, each solution is initialized with randomly selected genes or conditions as gene centers and condition centers, respectively. Suppose, the archive is initialized with 30 such randomly created solutions. For the above given data matrix, suppose first two initialized random solutions are given below.

Solution 1: [$g1$ $g3$ $c2$ $c4$]
Solution 2: [$g3$ $g6$ $g8$ $c3$ $c4$]

Here, in the first solution there are two gene cluster centers, viz. $g1$ and $g3$, and two condition cluster centers, viz. $c2$ and $c4$, which are picked up randomly from input set of genes and conditions, respectively. The length of the first solution therefore is 4. Similarly, second solution and rest 28 solutions are initialized according to Sect. 3.1.

Step 2 Assignment of points and updation of cluster centers: After 30 solutions are initialized as done in step 1, the rest data points are assigned to different gene/condition clusters and updation of cluster centers is also performed for each of them. The assignment is done using any one of three chosen distances (Euclidean, point symmetry and line symmetry). The chosen distance is utilized for assigning points for all of 30 solutions. For example: For Solution 1, rest genes are $g2$, $g4$, $g5$, $g6$, $g7$, $g8$ and $g9$. For each of them, their distances from two gene cluster centers which are $g1$ and $g3$ are calculated. Suppose we chose Euclidean distance for this purpose. They are calculated as,

$$d_e(g2, g1) = \sqrt{(2-1)^2 + (1-2)^2 + (3-1)^2 + (4-3)^2} = 2.64$$
$$d_e(g2, g3) = \sqrt{(2-1)^2 + (1-3)^2 + (3-4)^2 + (4-1)^2} = 3.87.$$

As, $d_e(g2, g1) < d_e(g2, g3)$ therefore, $g2$ is assigned to first cluster having center $g1$ of Solution 1. In the similar way, other genes are assigned to any one of two clusters. Finally, the following partitions are obtained for Solution 1:

$(g2, g4, g5, g7, g8) \in$ Gene cluster with center $g1$.
$(g6, g9) \in$ Gene cluster with center $g3$.

Similarly after assigning conditions to different condition clusters of Solution 1, the following clusters are obtained:

$c1 \in$ Condition cluster with center $c2$.
$c3 \in$ Condition cluster with center $c4$.

Once all genes/conditions are assigned to their respective clusters, then for each gene and condition clusters, the centers are updated as done in Sect. 3.2. The obtained first gene cluster in Solution 1 is as follows.

$$Clust1 = \begin{bmatrix} & c1 & c2 & c3 & c4 \\ g1 & 1 & 2 & 1 & 3 \\ g2 & 2 & 1 & 3 & 4 \\ g4 & 1 & 2 & 2 & 3 \\ g5 & 2 & 1 & 3 & 4 \\ g7 & 1 & 3 & 1 & 1 \\ g8 & 2 & 3 & 2 & 2 \end{bmatrix}$$

The present center of this cluster is $g1$ or (1, 2, 1, 3). After updation, the center becomes ($\frac{1+2+1+2+1+2}{6}$, $\frac{2+1+2+1+3+3}{6}$, $\frac{1+3+2+3+1+2}{6}$, $\frac{3+4+3+4+1+2}{6}$), i.e., (1.5, 2, 2, 2.83). Suppose we named the new gene cluster center for first cluster as g_{new1}. Similarly, the new centers of other gene and condition clusters of solution 1 are obtained as follows.

- Second Gene cluster previously having $g3$ or (1, 3, 4, 1) as center gets updated by (3, 1.66, 3.33. 1.33) named as g_{new2}.
- First Condition cluster previously having $c2$ or (2, 1, 3, 2, 1, 1, 3, 3, 1) as center gets updated by (1.5, 1.5, 2, 1.5, 1.5, 3, 2, 2.5, 2) named as c_{new1}.
- Second Condition cluster previously having $c4$ or (3, 4, 1, 3, 4, 1, 1, 2, 2) as center gets updated by (2, 3.5, 2.5, 2.5, 3.5, 2, 1, 2, 2.5) named as c_{new2}.

Similarly, for all 29 other solutions assignment of points (genes/conditions) and updation of centers are also performed. In place of Euclidean distance, point or line symmetry-based distance also can be used for assignment.

Step 3 Generating bi-clusters from solutions: After assigning points and updating centers of each gene/condition clusters, Solution 1 now can be represented as,

[g_{new1} g_{new2} c_{new1} c_{new2}]. The bi-clusters encoded within it are decoded according to Sect. 3.3 as follows.

Bi-cluster 1: $< g_{\text{new1}}\ c_{\text{new1}} >$
Bi-cluster 2: $< g_{\text{new1}}\ c_{\text{new2}} >$
Bi-cluster 3: $< g_{\text{new2}}\ c_{\text{new1}} >$
Bi-cluster 4: $< g_{\text{new2}}\ c_{\text{new2}} >$

Table 1 Comparative complexity analysis of different bi-clustering algorithms

Algorithms	Time complexity
MOPSOB	$O(\text{popsize} \times M_{\text{obj}} \times n^2)$
MOGAB	$O(\text{popsize} \times \text{TotalIter} \times M_{\text{obj}} \times n^2)$
SGAB	$O(\text{popsize} \times \text{TotalIter} \times \text{popsize} \times \text{mn})$
CC	$O((n+m) \times n \times m)$
RWB	$\text{TotalIter} \times \text{Cu} \times [(1-p) \times (n+m) + p]$
Bimax	$O(n \times m \times \beta \times \log \beta)$
OPSM	$O(n \times m^3 \times \text{TotalIter})$
ISA	xxx
BiVisu	$O(m^2 \times n \times \log m)$
Bi-MOCK	$n \times \log n$
HMOBI	xxx
SMOB	xxx
RGCE-B	xxx
LS-AMOSAB	If $mn \leq$ popsize, Overall complexity: $O(\text{TotalIter}/(\text{SL}-\text{HL}) \times \text{popsize}^2 \times \log(\text{popsize}))$ but if popsize $\leq mn$, Overall complexity becomes: $O(\text{popsize} \times mn)$

Each of other 29 solutions are decoded similarly.

Step 4 Calculation of objective functions: After decoding bi-clusters from each solution, their corresponding MSR and RV values are calculated according to Sect. 2.1 and 2.2. δ bi-clusters are only considered. For each solution its MSR is calculated by taking average of all MSR values of all δ bi-clusters within that solution. Then the front of non-dominated solutions is formed. Suppose the obtained MSR and RV of some solutions are as follows.

$$\begin{aligned}&\text{Solution 1: } (\text{MSR} = x,\ \text{RV} = y + \gamma)\\ &\text{Solution 2: } (\text{MSR} = x + \alpha,\ \text{RV} = y)\\ &\text{Solution 3: } (\text{MSR} = x - \beta,\ \text{RV} = y)\end{aligned}$$

where α, β, γ all are positive numbers. We can see that Solution 1 is superior to Solution 2 as its both MSR and RV values are better than corresponding MSR or RV values of Solution 2. So, Solution 1 dominates Solution 2. But if we compare between Solution 1 and Solution 3, we can see that Solution 1 is better than Solution 3 with respect to RV, but Solution 3 is better than Solution 1 with respect to MSR. So, Solution 1 and Solution 3 are non-dominating to each other.

In this way, the set of all non-dominated solutions are formed.

Step 5 Search operators: According to Sect. 3.5 each solution goes through one of three proposed mutation techniques. Please note that while a mutation operation is applied on a string or solution, it is applied either on gene cluster centers or on condition cluster centers at a time. Suppose, Solution 1, i.e., $[g_{\text{new1}}\ \ g_{\text{new2}}\ \ c_{\text{new1}}\ \ c_{\text{new2}}]$ goes through third mutation type, i.e., increasing the length of the string by 1. Let us assume, randomly a gene $g5$ from input set of genes is chosen and it is appended to existing string of the solution as a center of new gene cluster. The mutated Solution 1 becomes $[g_{\text{new1}}\ \ g_{\text{new2}}\ \ g5\ \ c_{\text{new1}}\ \ c_{\text{new2}}]$ with length 5. After applying mutation, again steps 2, 3, 4 are repeated until the stopping criteria are reached.

Step 6 Selecting best bi-clustering solution: Once the stopping criteria are reached, a final Pareto front is obtained. All non-dominating solutions are stored in the Pareto front. According to Sect. 3.6 if no information is known, then any one of these solutions can be chosen as the final solution. In this work, we have chosen the best solution with respect to lowest average MSR value of the solution.

4 Complexity analysis of the proposed bi-clustering approach using AMOSA

Let us assume that n be the number of points and m be the number of conditions, MaxLen be the maximum length of gene clusters, MaxLenC be the maximum number of condition clusters, TotalIter be the total number of generations, SL be the soft limit of the archive, popsize be the size of population, tolIndex be the total number of objective functions. As described in Bandyopadhyay et al. (2008) the basic operations and their worst case complexities are:

1. Archive initialization: $O(\text{SL})$
2. Compute membership functions and update the centers: If we are using Euclidean distance, then the complexity of this step is:

$$\begin{aligned}&O(\text{SL} \times (n \times \text{MaxLen} + m \times \text{MaxLenC}))\\ &\quad \approx O(\text{SL} \times (n+m))\end{aligned}$$

   If point or line symmetry-based distance is used, then it becomes:

$$\begin{aligned}&O(\text{SL} \times (n \times \text{MaxLen} \times \log n + m \times \text{MaxLenC} \times \log m))\\ &\quad \approx O(\text{SL} \times (n \log n + m \log m)).\end{aligned}$$

3. Forming non-dominated archive: $O(mn \times \text{SL} + 2\text{SL}^2)$.
4. Single Linkage Clustering: $O(\text{SL}^2 \times \log(\text{SL}))$ (Toussaint 1980). Clustering is carried out in the following situations:
   - Once after initialization if |archive| > HL, where |archive| is the size of archive
   - After each (SL–HL) number of iterations;
   - At the end if final |archive| > HL;

Table 2 Popular metrics of bi-clusters obtained by different algorithms on *Yeast dataset*

Algorithm	MSR		RV		BI index		Volume	
	Average	Best (min)	Average	Best (max)	Average	Best (min)	Average	Best (max)
LS-AMOSAB	155.23	65.67	**1789.53**	**5324.76**	**0.1496**	0.015	1954.76	7764
PS-AMOSAB	164.08	73.00	1673.81	4700.94	0.1785	0.1486	1836.56	6118
Eucli-AMOSAB	184.71	113.20	1576.05	3747.56	0.2428	0.1934	1765.92	3218
MOPSOB	218.54	200.58	789.85	1254.56	0.28	0.16	10510.8	15613
MOGAB	185.85	116.34	932.04	3823.46	0.3329	0.2123	329.93	1848
SGAB	198.88	138.75	638.23	3605.93	0.4026	0.2714	320.72	1694
CC	204.29	163.94	801.27	3726.22	0.3822	0.0756	1576.98	10523
RWB	295.81	231.28	528.97	1044.37	0.5869	0.2788	1044.43	5280
Bimax	**32.16**	**5.73**	39.53	80.42	0.4600	0.2104	60.52	256
OPSM	320.39	118.53	1083.24	3804.56	0.3962	**0.0012**	1533.31	3976
ISA	281.59	125.76	409.29	1252.34	0.7812	0.1235	79.22	168
BiVisu	290.59	240.96	390.73	775.41	0.7770	0.3940	2136.34	4080
Bi-MOCK	203.56	–	1307.42	–	0.16	–	15530.19	–
HMOBI	299.6	–	789.12	–	0.38	–	7827.86	–
SMOB	206.17	–	678.53	–	0.3	–	415.06	–
RGCE-B	181.8	–	–	–	–	–	1956.2	–

Table 3 Popular metrics of bi-clusters obtained by different algorithms on *Human dataset*

Algorithms	MSR		RV		BI Index		Volume	
	Average	Best (min)	Average	Best (max)	Average	Best (min)	Average	Best (max)
LS-AMOSAB	**67.49**	61.75	**10459.37**	**15956.498**	**0.105**	**0.004**	3967.52	10864
PS-AMOSAB	73.34	55.47	9958.86	10372.27	0.129	0.119	2411.96	9296
Eucli-AMOSAB	273.69	92.52	9220.18	10004.54	0.145	0.132	1117.80	6398
MOPSOB	927.47	745.25	4348.2	8745.65	0.213	0.09	34012.24	37666
MOGAB	801.37	569.23	2378.25	5377.51	0.4710	0.2283	276.48	1036
SGAB	855.36	572.54	2222.19	5301.83	0.5208	0.3564	242.93	996
CC	1078.38	779.71	2144.14	5295.42	0.5662	0.2298	388.86	9100
RWB	1185.69	992.76	1698.99	3575.40	0.7227	0.3386	851.54	1830
Bimax	387.71	96.98	670.83	3204.35	0.7120	0.1402	64.34	138
OPSM	1249.73	**43.07**	6374.64	11854.63	0.2520	0.1024	373.5472	1299
ISA	2006.83	245.28	4780.65	14682.47	0.6300	0.0853	69.35	220
BiVisu	1680.23	1553.43	1913.24	2468.63	0.8814	0.6537	1350.24	17739
Bi-MOCK	821.93	–	4113.05	–	0.2	–	75637.88	–
HMOBI	1199.9	–	2231.04	–	0.54	–	47356.56	–
SMOB	1019.16	–	3169.19	–	0.32	–	505.41	–

Therefore, the maximum number of times the clustering procedure can be called: (TotalIter/(SL−HL))+2 Therefore, the total complexity due to clustering procedure is $O((\text{TotalIter}/(\text{SL}-\text{HL})) \times \text{SL}^2 \times \log(\text{SL}))$.

5. Checking domination status and considering one of three cases of AMOSA: $O(\text{TotalIter} \times (mn + 2\text{SL}))$.

As third and fourth components have more impact than first, second and fifth components, during further analysis we have considered only third and fourth components only. Therefore, the overall complexity becomes,

$$O(mn \times \text{SL} + 2\text{SL}^2 + (\text{TotalIter}/(\text{SL}-\text{HL})) \times SL^2 \times \log(\text{SL})) \approx O(mn \times \text{SL} + (F \times \text{SL}^2 \times \log(\text{SL}))).$$

where $F = \text{TotalIter}/(\text{SL}-\text{HL})$. If $mn \leq \text{SL}$, overall complexity becomes: $O(F \times \text{SL}^2 \times \log(\text{SL}))$, but if $\text{SL} \leq mn$, overall complexity becomes: $O(\text{SL} \times mn)$. Initially, the value of SL = popsize.

Table 4 Popular metrics of bi-clusters obtained by different algorithms on *Leukemia dataset*

Algorithms	MSR		RV		BI index		Volume	
	Average	Best (min)	Average	Best (max)	Average	Best (min)	Average	Best (max)
LS-AMOSAB	**87.48**	**18.66**	8267.23	17896.74	**0.1217**	**0.0024**	592	3964
PS-AMOSAB	98.67	19.65	3858.61	14629.23	0.1385	0.1273	498	2496
Eucli-AMOSAB	102.43	20.81	2890.43	12785.91	0.1586	0.1474	513.21	2605
MOGAB	206.55	59.84	2852.76	27688.93	0.1698	0.0553	330.56	1008
SGAB	243.54	79.41	2646.25	21483.74	0.2663	0.0803	323.54	840
CC	335.81	109.49	2794.36	24122.64	0.2648	0.0071	230.90	1124
RWB	583.42	561.73	1137.62	2937.76	0.4156	0.1123	388.65	420
Bimax	1243.45	703.45	1834.54	13842.65	0.3475	0.0694	95.25	130
OPSM	3.58E6	5.34E4	**5.03E6**	1.73E7	0.6375	0.1359	3176.34	12226
ISA	1 .12E7	1.03E6	1.13E7	**3.18E7**	0.7638	0.7556	731.86	1398
BiVisu	1580.45	587.40	3006.15	12547.74	0.4540	0.1018	2380.13	3072

Comparative complexity analysis with other algorithms We have compared the time complexity of our proposed bi-clustering algorithm with other existing techniques, and those are shown in Table 1. Please note that we have reported the complexities of other existing techniques as reported in the existing literature. Complexities of some algorithms are not available. We have indicated those missing fields as 'xxx'. In Table 1,

n: number of genes.
m: number of conditions.
β: number of bi-clusters obtained from input dataset.
TotalIter: total number of generations.
p: probability of random move.
Cu: computing new residue.
popsize: size of population.
M_{obj}: number of objective functions.

From the table it can be seen that the average time complexity of our proposed bi-clustering algorithm is comparable to other existing evolutionary bi-clustering techniques.

5 Experimental results

All versions of our proposed AMOSA-based bi-clustering algorithm are implemented in C on an Intel core i5, 2.20 GHz. Windows 8.1 machine.

5.1 Chosen datasets

Three datasets are chosen for experiment which are *Yeast* (Cheng and Church 2000), *Human Large B Cell Lymphoma* dataset (Cheng and Church 2000) and *Acute Lymphoblastic Leukemia (ALL)* data (Golub et al. 1999). *Yeast* data collect expression levels of 2884 genes under 17 conditions, *Human lymphoma* dataset contains 4026 genes and 96 conditions, and *Leukemia* data contain 7129 probes of human genes with 47 ALL samples and 25 AML samples. *Yeast* and *Human* datasets are publicly available from http://arep.med.harvard.edu/bi-clustering Web site. The *Leukemia* data can be downloaded from http://sdmc.lit.org.sg/GEDatasets/Datasets.html.

The values of δ for the above-mentioned datasets are 300, 1200 and 500, respectively. As we are comparing our approach with another multi-objective-based bi-clustering technique MOGAB (Maulik et al. 2009) and other bi-clustering techniques as reported in (Maulik et al. 2009; Liu et al. 2008), we have chosen δ values as the same as those selected in the previous approaches. The missing values are removed from the datasets to avoid any random interference. Also to make the datasets comparable to Maulik et al. (2009), these are normalized to zero mean and unit variance.

Please note that our proposed *PS-AMOSAB* and *LS-AMOSAB* are based on two symmetry-based distance measures, which are capable of identifying symmetric clusters. Therefore, if datasets contain some line symmetric bi-clusters, then it can be identified by *LS-AMOSAB* but not by *PS-AMOSAB* or *Eucli-AMOSAB*. If dataset contains any point symmetric cluster, then it can be identified by *PS-AMOSAB* and *LS-AMOSAB* but not by *Eucli-AMOSAB*. If dataset does not contain any symmetric clusters, then we cannot distinguish between *PS-AMOSAB*, *LS-AMOSAB* and *Eucli-AMOSAB* from efficiency perspective. Therefore, all the three versions of our proposed bi-clustering technique can be applied on any kind of dataset, but symmetry distance-based versions will distinctly work well for datasets possessing some symmetric property.

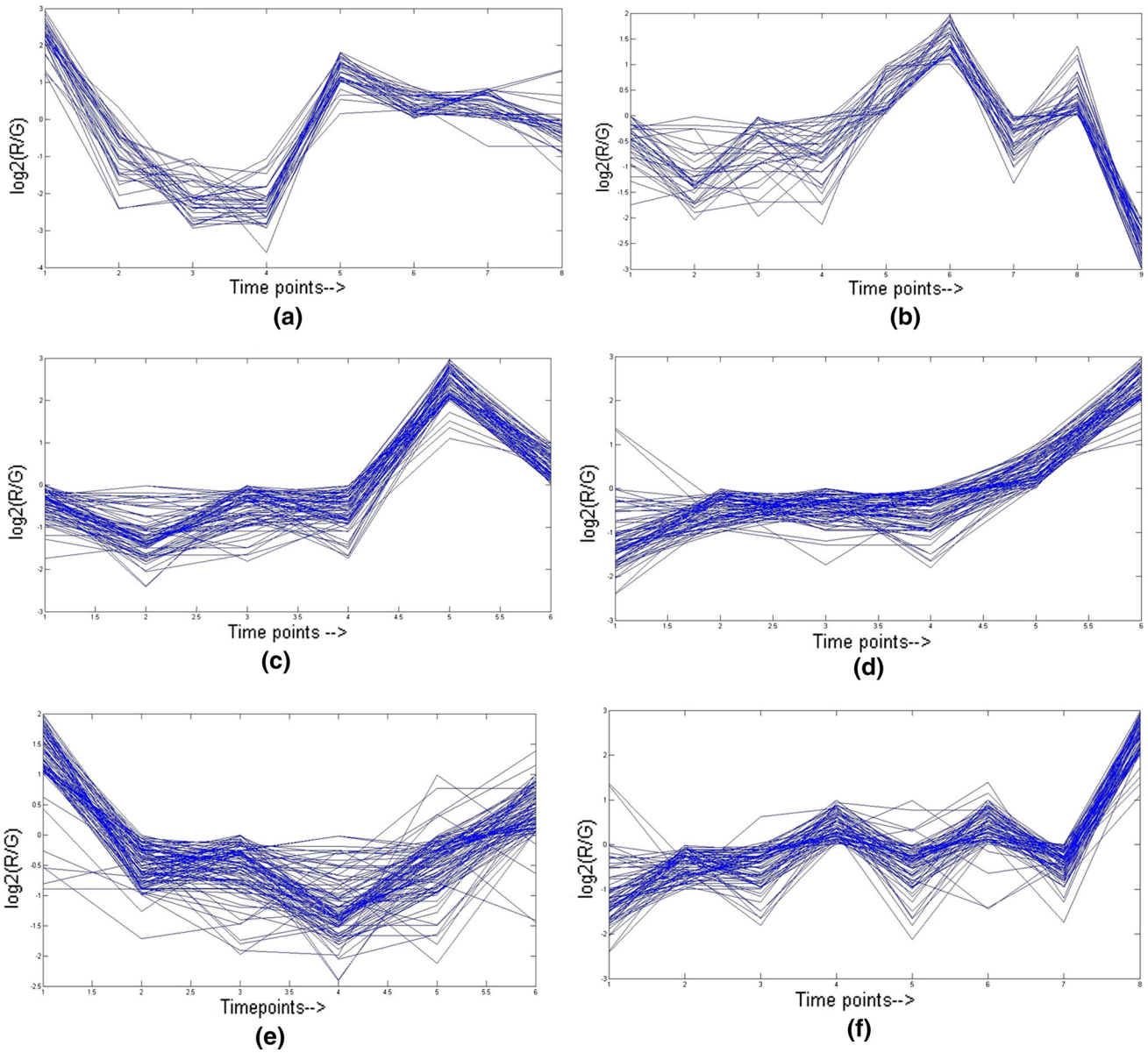

Fig. 8 Six bi-clusters of *Yeast* data determined by *LS-AMOSAB*. For each bi-cluster, number of genes, number of conditions, MSR, variance, BI are given as follows. **a** 43, 8, 65.67, 4500, 0.015. **b** 47, 9, 98.567, 4209.56, 0.0234. **c** 59, 6, 110.985, 3956.456, 0.028. **d** 58, 6, 109.8871, 3882.796, 0.0282. **e** 84, 6, 149.543, 5324.76, 0.028. **f** 81, 6, 92.35, 5100.4, 0.018

5.2 Input parameters

We have executed AMOSA-based bi-clustering technique with the following parameter combinations:

$T_{\min} = 0.0001$, $T_{\max} = 100$, $\alpha = 0.9$, HL $= 50$, SL $= 100$ and TotalIter $= 100$.

After a thorough sensitivity study, the following parameter values were chosen. The three main parameters which we have selected by sensitivity study are,

(i) initial value of temperature ($T_{\max}$),
(ii) cooling schedule,
(iii) number of iterations to be performed at each temperature.

It has been mentioned in Bandyopadhyay et al. (2008) that the temperature must lie in the mentioned range in order to allow the simulated annealing to perform random walk over the landscape. As done in Bandyopadhyay et al. (2008), we have also set the initial temperature to achieve an initial acceptance rate of approximately 50% on derogatory proposals. The geometrical cooling schedule α is chosen in the range between 0.5 and 0.99 according to Bandyopad-

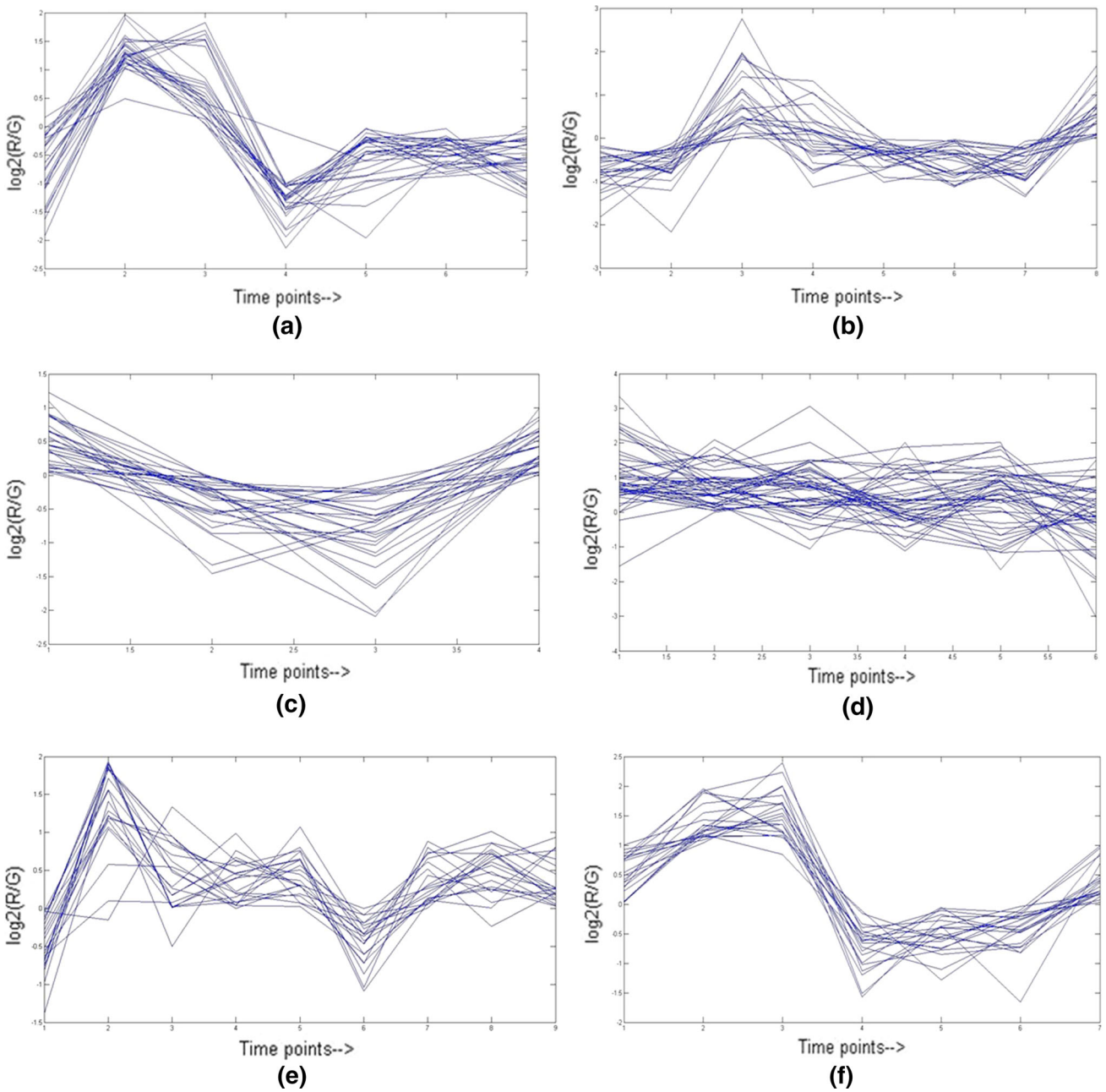

Fig. 9 Six bi-clusters of *Human* data determined by *LS-AMOSAB*. For each bi-cluster, number of genes, number of conditions, MSR, variance, BI index values are given. **a** 27, 7, 61.75, 13765.27, 0.0045. **b** 24, 8, 67.675, 10675.43, 0.0063. **c** 28, 4, 78.683, 11686.4, 0.0067. **d** 46, 6, 79.9673, 9693.646, 0.0082. **e** 19, 9, 92.573, 15956.498, 0.0058. **f** 24, 7, 61.96, 15490.4, 0.004

hyay et al. (2008). We have varied the value of α in this range by keeping other parameters constant. Finally the value of α corresponding to the best MSR value of the obtained solution is selected as the final value. In order to make the system sufficiently close to the stationary distribution, the number of iterations is chosen as 100. We observed that the MSR value of resulting solution gets saturated even if we increased the value of TotalIter. So it is best to keep TotalIter = 100.

To make our results consistent for all the datasets the above-mentioned parameters were chosen. The above-mentioned sensitivity study does not require any estimation strategy.

5.3 Chosen validity indices

In order to select the final bi-clustering solution for each version of our proposed algorithms we have considered average

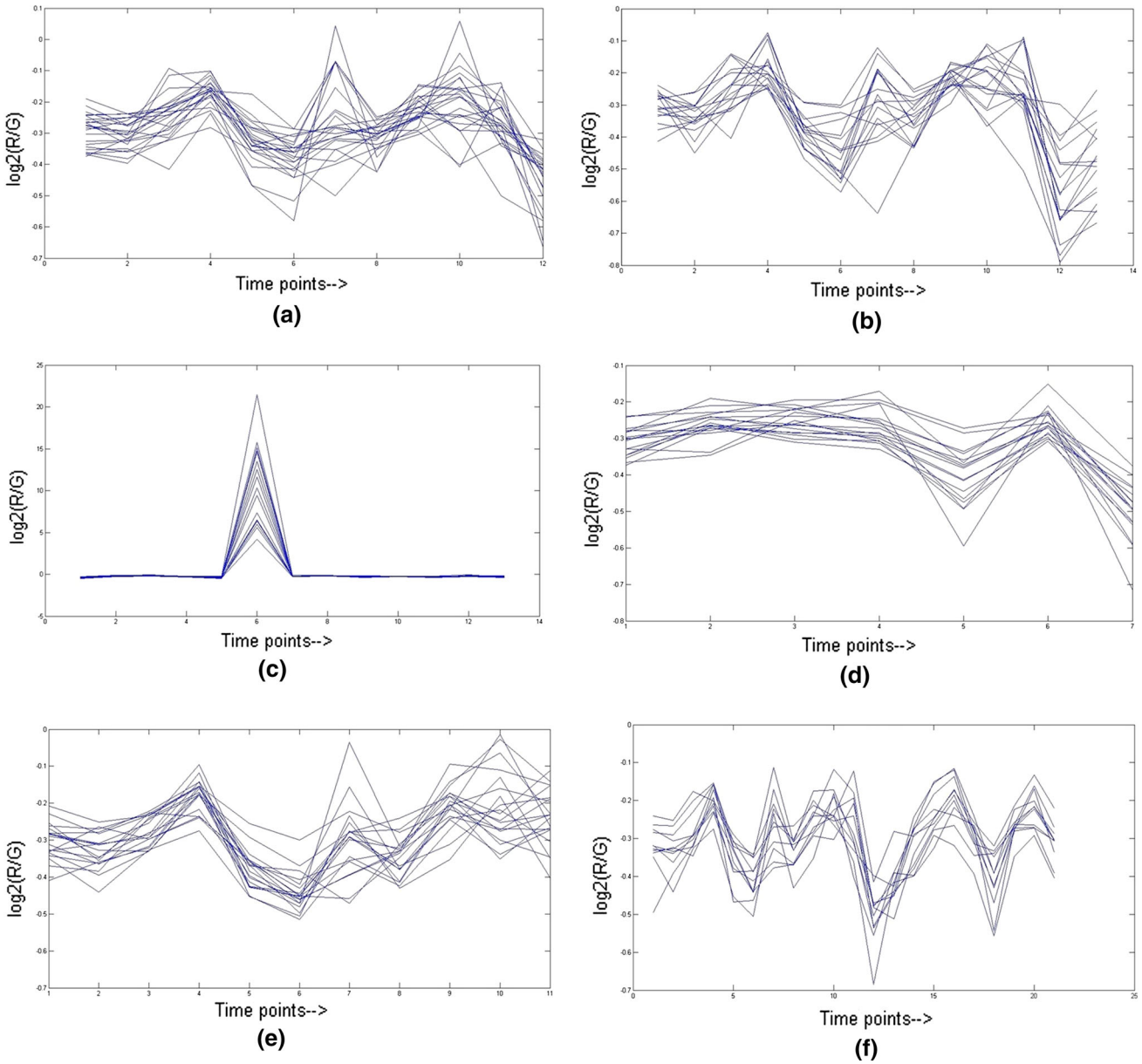

Fig. 10 Six bi-clusters of *Leukemia* data determined by *LS-AMOSAB*. For each bi-cluster, number of genes, number of conditions, MSR, var, BI index values are given. **a** 23, 12, 18.66, 7856.85, 0.0024. **b** 17, 13, 78.0423, 9983.23, 0.0078. **c** 16, 13, 88.7364, 13468.4278, 0.0066. **d** 15, 7, 92.76, 12056.6, 0.0076. **e** 19, 11, 63.86, 11058.8, 0.0058. **f** 11, 21, 82.0384, 12056.6, 0.0068

and best MSR (Cheng and Church 2000) and RV (Cheng and Church 2000) values. Lower the value of MSR score indicates larger coherence, and better is the richness of the bi-cluster. High RV is more interesting because this makes significant changes in the expression levels of the genes. We have also chosen BI score as cluster quality measure. It is defined as follows.

For a bi-cluster having MSR value H and RV value R the BI value will be calculated as $\text{BI} = \frac{H}{(1+R)}$ non-triviality and high coherence of a bi-cluster are represented by lower BI values.

5.4 Comparing approaches

We have also compared our proposed three versions of bi-clustering techniques i.e., *Eucli-AMOSAB*, *PS-AMOSAB* and *LS-AMOSAB*, with some other state-of-the-art single-objective and multi-objective bi-clustering techniques which are Bi-MOCK (Bousselmi et al. 2017), HMOBI (Seridi et al. 2015), SMOB (Divina and Aguilar-Ruiz 2007), RGCE-B (Yan and Wang 2013), MOPSOB (Liu et al. 2008), MOGAB (Maulik et al. 2009), Cheng and Church (CC) (Cheng and Church 2000), RWB (Angiulli and Pizzuti 2005), OPSM

Table 5 Results for biological significance test: the best six biologically enriched (from the aspect of functionality) bi-clusters produced by our proposed *LS-AMOSAB* on *Yeast* data

Bi-clusters	GO term	Cluster %	Genome %
Bi-cluster 1	GO:0022625	17.1	10.4
	cytosolic large ribosomal subunit		
	GO:0042221	10.43	8.39
	response to chemical		
	GO:0006325	18.62	12.35
	chromatin organization		
	GO:0006974	15.62	9.30
	cellular response to DNA damage stimulus		
	GO:0055085	17.94	8.32
	transmembrane transport		
Bi-cluster 2	GO:0015934	14.1	12.1
	large ribosomal subunit		
	GO:0006366	16.74	8.43
	transcription from		
	RNA polymerase II promoter		
	GO:0006811	17.29	9.39
	ion transport		
Bi-cluster 3	GO:0044699	21.1	15.3
	single-organism process		
	GO:0044763	23.2	15.0
	single-organism cellular process		
	GO:0051179	25.1	18.8
	localization		
	GO:0051641	15.9	11.4
	cellular localization		
	GO:0051234	21.8	16.7
	establishment of localization		
	GO:0033036	13.7	9.6
	macromolecule localization		
	GO:1902578	12.7	8.8
	single-organism localization		
Bi-cluster 4	GO:0000788	17.2	10.1
	nuclear nucleosome		
	GO:0000786	16.2	10.2
	nucleosome		
	GO:0005657	21.7	15.8
	replication fork		
Bi-cluster 5	GO:0000278	18.63	10.77
	mitotic cell cycle		
	GO:0023052	18.45	12.75
	signaling		
	GO:0005975	19.92	10.9
	carbohydrate metabolic process		
	GO:0048285	18.75	14.62
	organelle fission		

Table 5 continued

Bi-clusters	GO term	Cluster %	Genome %
Bi-cluster 6	GO:0006281 DNA repair	19.2	13.4
	GO:0006629 lipid metabolic process	15.94	9.32
	GO:0048285 organelle fission	19.78	12.62

(Ben-Dor et al. 2003), ISA (Ihmels et al. 2004), BiVisu (Cheng et al. 2007), SGAB (Tanay et al. 2002), Bimax (Hartigan 1972).

5.5 Discussion of obtained results

We have executed all of the three versions of our proposed bi-clustering algorithm, i.e., *Eucli-AMOSAB*, *PS-AMOSAB* and *LS-AMOSAB*, on all three chosen datasets and compared the outcomes with existing state-of-the-art bi-clustering algorithms with respect to average and best (minimum or maximum) values of MSR, RV and BI score. The obtained results are reported in Tables 2, 3 and 4 for *Yeast*, *Human* and *Leukemia* datasets, respectively. Along with values of chosen validity measures we have also reported average and maximum volumes of bi-clustering solution. In Figs. 8, 9 and 10, six best bi-clusters discovered by *LS-AMOSAB* algorithm for *Yeast*, *Human* and *Leukemia* data, respectively, are shown through cluster profile plot. These bi-clusters have low MSR values, high RV values and low BI scores. For each plot, points on X-axis represent time points or conditions and Y-axis indicates genes within the corresponding bi-cluster. For a particular condition on X-axis and gene on Y-axis the corresponding value on the plot represents the log normalized expression value of that gene for the condition. From these plots we can see that the obtained bi-clusters for each dataset are quite coherent or similar to each other with respect to the conditions within those bi-clusters. From Tables 2, 3 and 4 it can be seen that with respect to average and best MSR, RV and BI score all three proposed versions of bi-clustering algorithms perform better than most of the state-of-the-art (both single and multi-objective) algorithms for all three datasets.

It was also very prominently observed that all three versions perform better than other MOO-based bi-clustering technique, Bi-MOCK (Bousselmi et al. 2017), HMOBI (Seridi et al. 2015), MOGAB (Maulik et al. 2009) and MOPSOB (Liu et al. 2008) for almost all three indices (MSR, RV, BI). If we compare between three versions of our proposed algorithm, then we can observe that for all three indices, *LS-AMOSAB* performs better than *PS-AMOSAB* and *PS-AMOSAB* performs better than *Eucli-AMOSAB*. The best values are highlighted in bold font in each table. For MOPSOB (Liu et al. 2008), Bi-MOCK (Bousselmi et al. 2017), HMOBI (Seridi et al. 2015), SMOB (Divina and Aguilar-Ruiz 2007) algorithms, results are not available for *Leukemia* data. For RGCE-B (Yan and Wang 2013) algorithm results for both *Leukemia* and *Human* data sets are not available. Also for these algorithms, best values of other indices are not available for other datasets. In Tables 2 and 3 the missing values are represented by '–'. The obtained results prove the following,

1. Symmetry-based distances perform better than traditional distance measure to obtain good-quality coherent bi-clusters from a given dataset.
2. Line symmetry-based distance performs best among all the three chosen distance measures.

5.6 Biological significance test

In Tables 2, 3 and 4, the superiority of *LS-AMOSAB* has been shown over other state-of-the-art bi-clustering techniques as well as our other proposed versions of bi-clustering algorithms. Now in order to prove that bi-clusters obtained by *LS-AMOSAB* are biologically significant, we have conducted biological significance test. The bi-clusters obtained for *Yeast* dataset were tabbed for this test with the help of GOTermFinder (http://db.yeastgenome.org/cgi-bin/GO/goTermFinder). In Table 5 we have summarized significant GO terms shared by genes of all six mostly enriched bi-clusters of best solution with respect to BI index. Instead of BI index, MSR or RV measures also could have chosen. For each GO term, the percentage of genes sharing that term among the genes of that bi-cluster and among the whole genome have been reported. Results clearly signify that genes of the same bi-cluster share the higher percentage of GO terms compared to the whole genome. This indicates that the genes of a particular bi-cluster are more involved in similar biological processes compared to the remaining genes of the genome.

5.7 Statistical significance test

To prove the supremacy of *LS-AMOSAB* statistically, we have conducted a statistical significance test (Sirkin 2005)

Table 6 *p* values of *t* tests comparing the mean BI scores of bi-clusters of our proposed *LS-AMOSAB* with respect to other existing algorithms

Algorithms	*p* value		
	Yeast	*Human*	*Leukemia*
PS-AMOSAB	1.65E−216	1.46E−187	2.874E−235
Eucli-AMOSAB	2.785E−297	2.64E−263	3.1443E−219
MOPSOB	3.96E−321	3.279E−265	–
MOGAB	6.18E−278	2.26E−31	1.478E−19
SGAB	3.95E−276	2.46E−32	1.57E−27
CC	5.23E−275	3.85E−33	2.19E−27
RWB	2.55E−288	2.75E−35	2.134E−27
Bimax	1.86E−278	3.97E−35	1.314E−27
OPSM	7.36E−276	5.875E−26	1.478E−28
ISA	5.92E−292	4.272E−34	1.337E−28
BiVisu	6.38E−222	4.27E−37	2.966E−23
Bi-MOCK	3.45E−182	2.956E−95	–
HMOBI	3.56E−123	4.23E−95	–
SMOB	4.23E−85	3.45E−213	–

named as *t* test at 5% significance level. As null hypothesis, we assume that there are insignificant differences between mean values of two groups. According to alternative hypothesis there are significant differences in the mean values of two groups. Fourteen groups corresponding to fourteen algorithms (all chosen and proposed bi-clustering algorithms except *LS-AMOSAB* and RGCE-B) are created. The RV value of RGCE-B (Yan and Wang 2013) algorithm is not available. So, we could not compute corresponding BI score. Between each pair of groups *t* test is performed based on obtained BI score and corresponding *p* value is reported. As for *Leukemia* dataset results of some algorithms are not available; therefore, we have denoted those field by '–'. From Table 6 it can be seen that the *p* value in each case is less than 0.05. This outcome supports the alternative hypothesis i.e., the supremacy of *LS-AMOSAB* over other proposed and existing bi-clustering approaches.

6 Conclusions and future works

In this scholarly work, we have proposed a MOO-based bi-clustering technique aided by optimization technique AMOSA (Bandyopadhyay et al. 2008) as the underlying optimization strategy. A case study on the suitability of different distance measures including Euclidean, point symmetry and Line symmetry-based distance in solving the problem of bi-clustering is also conducted. From our obtained results, we can conclude that all the three versions of our proposed bi-clustering algorithm utilizing different distance measures perform better than most of the state-of-the-art algorithms for all three datasets with respect to MSR, RV and BI performance measures. Results also reveal that *PS-AMOSAB* performs better than *Eucli-AMOSAB*. Also it reveals that *LS-AMOSAB* performs better than *PS-AMOSAB*. The conducted biological and statistical significance tests further support the obtained results. In future, we would like to explore a new paradigm of clustering, namely tri-clustering. Tri-clustering goes one step further of bi-clustering and aims to concurrently cluster two data tables that share a common set of row labels, but whose column labels are distinct (Zhao and Zaki 2005). The concepts of tri-clustering can be applied in the field of microarray gene classification.

Acknowledgements The first author sincerely thanks Tata Consultancy Services (TCS) for providing funding to conduct this research.

Compliance with ethical standards

Conflict of interest Authors Sudipta Acharya, Sriparna Saha and Pracheta Sahoo declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Not applicable.

References

Acharya S, Saha S (2016) Importance of proximity measures in clustering of cancer and mirna datasets: proposal of an automated framework. Mol BioSyst 12(11):3478–3501

Acharya S, Saha S, Thadisina Y (2016) Multiobjective simulated annealing-based clustering of tissue samples for cancer diagnosis. IEEE J Biomed Health Inf 20(2):691–698

Angiulli F, Pizzuti C (2005) Gene expression biclustering using random walk strategies. In: International conference on data warehousing and knowledge discovery. Springer, pp 509–519

Attneave F (1955) Symmetry, information, and memory for patterns. Am J Psychol 68(2):209–222

Bandyopadhyay S, Saha S (2007) Gaps: A clustering method using a new point symmetry-based distance measure. Pattern Recogn 40(12):3430–3451

Bandyopadhyay S, Saha S (2012) Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications. Springer, Berlin

Bandyopadhyay S, Saha S, Maulik U, Deb K (2008) A simulated annealing-based multiobjective optimization algorithm: Amosa. IEEE Trans Evol Comput 12(3):269–283

Ben-Dor A, Chor B, Karp R, Yakhini Z (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. J Comput Biol 10(3–4):373–384

Bousselmi M, Bechikh S, Hung C-C, Said LB (2017) Bi-mock: a multi-objective evolutionary algorithm for bi-clustering with automatic determination of the number of bi-clusters. In: International conference on neural information processing. Springer, pp 366–376

Bryan K, Cunningham P, Bolshakova N (2005) Biclustering of expression data using simulated annealing. In: 18th IEEE symposium on

computer-based medical systems, 2005. Proceedings. IEEE, pp 383–388
Chakraborty A, Maka H (2005) Biclustering of gene expression data using genetic algorithm. In: Proceedings of the 2005 IEEE symposium on computational intelligence in bioinformatics and computational biology, 2005. CIBCB'05. IEEE, pp 1–8
Cheng K-O, Law N-F, Siu W-C, Lau T (2007) Bivisu: software tool for bicluster detection and visualization. Bioinformatics 23(17):2342–2344
Cheng Y, Church GM (2000) Biclustering of expression data. Ismb 8(2000):93–103
Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans Evol Comput 6(2):182–197
Deb K, Sindhya K, Hakanen J (2016) Multi-objective optimization. In: Decision sciences: theory and practice. CRC Press, Boca Raton, FL
Divina F, Aguilar-Ruiz JS (2007) A multi-objective approach to discover biclusters in microarray data. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM, pp 385–392
Dudoit S, Fridlyand J (2003) Classification in microarray experiments. Stat Anal Gene Expr Microarray Data 1:93–158
Getz G, Levine E, Domany E (2000) Coupled two-way clustering analysis of gene microarray data. Proc Nat Acad Sci 97(22):12079–12084
Giancarlo R, Bosco GL, Pinello L (2010) Distance functions, clustering algorithms and microarray data analysis. In: International conference on learning and intelligent optimization. Springer, pp 125–138
Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA et al (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439):531–537
Hartigan JA (1972) Direct clustering of a data matrix. J Am Stat Assoc 67(337):123–129
Hochreiter S, Bodenhofer U, Heusel M, Mayr A, Mitterecker A, Kasim A, Khamiakova T, Van Sanden S, Lin D, Talloen W et al (2010) Fabia: factor analysis for bicluster acquisition. Bioinformatics 26(12):1520–1527
Huang Q, Tao D, Li X, Liew A (2012) Parallelized evolutionary learning for detection of biclusters in gene expression data. IEEE/ACM Trans Comput Biol Bioinf 9(2):560–570
Ihmels J, Bergmann S, Barkai N (2004) Defining transcription modules using large-scale gene expression data. Bioinformatics 20(13):1993–2003
Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Inc
Liu J, Li Z, Liu F, Chen Y (2008) Multi-objective particle swarm optimization biclustering of microarray data. In: IEEE international conference on bioinformatics and biomedicine, 2008. BIBM'08. IEEE, pp 363–366
Maulik U, Mukhopadhyay A, Bandyopadhyay S (2009) Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm. IEEE Trans Inf Technol Biomed 13(6):969
Ray SS, Bandyopadhyay S, Pal SK (2007) New distance measure for microarray gene expressions using linear dynamic range of photo multiplier tube. In: International conference on computing: theory and applications, 2007. ICCTA'07. IEEE, pp 337–341
Sahoo P, Acharya S, Saha S (2016) Automatic generation of biclusters from gene expression data using multi-objective simulated annealing approach. In: 2016 23rd international conference on pattern recognition (ICPR). IEEE, pp 2174–2179
Seifoddini HK (1989) Single linkage versus average linkage clustering in machine cells formation applications. Comput Ind Eng 16(3):419–426
Seridi K, Jourdan L, Talbi E-G (2015) Using multiobjective optimization for biclustering microarray data. Appl Soft Comput 33:239–249
Sirkin RM (2005) Statistics for the social sciences. Sage Publications
Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(suppl 1):S136–S144
Toussaint GT (1980) Pattern recognition and geometrical complexity. In: Proceedings of the 5th international conference on pattern recognition, vol 334, p 347
Yan D, Wang J (2013) Biclustering of gene expression data based on related genes and conditions extraction. Pattern Recogn 46(4):1170–1182
Yang J, Wang H, Wang W, Yu P (2003) Enhanced biclustering on expression data. In: 3rd IEEE symposium on bioinformatics and bioengineering, 2003. Proceedings. IEEE, pp 321–327
Zhang Z, Teo A, Ooi BC, Tan K-L (2004) Mining deterministic biclusters in gene expression data. In: 4th IEEE symposium on bioinformatics and bioengineering, 2004. BIBE 2004. Proceedings. IEEE, pp 283–290
Zhao L, Zaki MJ (2005) Tricluster: an effective algorithm for mining coherent clusters in 3d microarray data. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM, pp 694–705