# Optimizing the minimum spanning tree-based extracted clusters using evolution strategy

**Zahid Halim**[1] · **Uzma**[1]

Received: 10 February 2017 / Revised: 30 March 2017 / Accepted: 13 April 2017 / Published online: 26 April 2017

**Abstract** There are many approaches available for extracting clusters. A few are based on the partitioning of the data and others rely on extracting hierarchical structures. Graphs provide a convenient representation of entities having relationships. Clusters can be extracted from a graph-based structure using minimum spanning trees (MSTs). This work focuses on optimizing the MST-based extracted clusters using Evolution Strategy (ES). A graph may have multiple MSTs causing varying cluster formations based on different MST selection. This work uses (1+1)-ES to obtain the optimal MST-based clustering. The Davies–Bouldin Index is utilized as fitness function to evaluate the quality of the clusters formed by the ES population. The proposed approach is evaluated using eleven benchmark datasets. Seven of these are based on microarray and the rest are taken from the UCI machine learning repository. Both, external and internal cluster validation indices are used to evaluate the results. The performance of the proposed approach is compared with two state-of-the-art MST-based clustering algorithms. The results support promising performance of the proposed approach in terms of time and cluster validity indices.

**Keywords** Minimum spanning trees · Clustering · Graphs · Evolution strategy

## 1 Introduction

Clustering is a key machine learning and data mining task used for extracting useful patterns in diverse fields. Over the period of decades, various clustering techniques have been proposed [1]. Connectivity-based clustering, centroid-based clustering, distribution-based clustering, and density-based clustering are a few of the most commonly used approaches for cluster analysis. Clustering has many real-world applications. A few cases in point are extracting coherent groups from a protein–protein interaction network, wireless sensor network with uncertain edges, and social networking. Depending on the problem domain, cluster analysis is also focused on finding non-spherical clusters. Traditional partitioning-based clustering methods, like $k$-means and $k$-medoids fail to find non-spherical clusters [2]. Graph-based approaches have shown promising results for this task. Generally, the goal of clustering is to extract strongly connected, coherent groups from the underlying data. Graphs provide a convenient representation of items having a direct or indirect relation. Where related items in a graph are represented by an edge between nodes or a series of edges among scattered nodes. Spanning trees represent a structure that completely covers the graph having a path to every node. Minimum spanning trees (MSTs) are similar to the spanning tree, however, the overall sum of edges' weight in the MST is minimum. Two nodes closer to each other in a graph will have lesser value of the edge weight connecting them as compared to the nodes placed far apart. However, a different interpretation of an edge weight is also possible, specific to the problem domain. Keeping the distance-based interpretation in view, an MST thus represents a closely connected group of nodes that can be interpreted as a cluster. When utilizing MSTs for cluster formation, using a complete MST to represent a single cluster is not an appropriate concept [3]. Reason

✉ Zahid Halim
zahid.halim@giki.edu.pk

Uzma
uzma@giki.edu.pk

[1] Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan

being the inclusion of all nodes in the MST for a connected graph, thus forming a single cluster. Techniques employing MSTs for clustering address this concern by restricting growth of an MST for a specific threshold weight. This enables to construct coherent groups that are strongly connected. These MST-based extracted groups can further be optimized by rearranging their connections within the tree. However, this rearrangement needs to be done in line with the actual connections in the graph. It needs to be done gradually and evaluated for improvement in cluster quality before accepting it. This is an area where evolutionary computing can be utilized. Genetic algorithm and genetic programming can be controlled to converge steadily using mutation and crossover rates. However, evolution strategy (ES) seems to be a better choice here, where the reproduction operator of mutation is only used. An approach can be devised that extracts MST from a graph representing a cluster/pattern. Later, this MST can be optimized using ES-based approach. Since a graph may have multiple MSTs, these can become candidate solutions (population) for the ES. Previously, ES-based approaches for optimizing solutions in the domain of pattern recognition, independent learning, and bioinformatics have shown better results [4–6]. Clustering data using evolutionary algorithms has been studied before, however, most of the existing literature is either focused on similarity measures for data [7] or is an extension of existing algorithms for graphs [8]. There are a few existing methods that rely on the graph-based representation of the MST-based methodology for clustering [9], similar to the approach proposed in this work, but the amount of data lost in these methods is much higher [10,11]. MST-based clustering is known for extracting irregular boundaries and outlier detection [12]. The MST-based clustering techniques have widely been used for efficient clustering. However, creating a single MST from a graph is not a good idea for efficient clustering. To overcome this limitation this work proposes to optimize MSTs for efficient clustering formations. The main idea is to extract multiple MSTs from a graph and then optimize the clustering formations using ES. This work utilizes (1+1)-ES for the optimization of MST-based extracted clusters. Details on the existing approaches for clustering data and how the proposed approach is different from the existing work are listed in Sect. 2.

This work focuses on optimizing the MST-based clustering using the ES. A graph may have many MSTs and by selecting any particular MST different clustering formation can be achieved. In order to obtain the optimal MST-based clustering (1+1)-ES is used in this work. Mutation is the only reproduction operator used in ES for the creation of new individuals by modification of the parent solutions. The ES population is generated using multiple MSTs of the same input graph. Later, these MST-based clustering formations are optimized using the objective function. The Davies–Bouldin Index (DBI) is used as a fitness function to evaluate the quality of clusters formed by each individual of the ES population. This proposal can be adopted as a generic framework, where DBI is one of the many possible fitness functions to guide this framework for better clustering formations. The reason for opting for DBI here is its wide utility in the clustering algorithms and much less complex than the computation of other cluster validity indices like Silhouettes coefficient (SC). Additionally, DBI being an internal validity index enables to assess the cluster quality in the absence of the ground truth. Thus, justifying the utility of DBI as an objective function here. The ES is executed for 1000 iterations having a population of ten individuals. Population size of ten individuals is opted due to the limited number of disjoint MSTs in a graph. There can be a case where the input graph having all edges with same weight has only a single MST for any root node selection. The proposed approach is evaluated using eleven benchmark datasets. Seven of these are based on microarray and the rest are taken from the UCI machine learning repository. This evaluation is performed using the external cluster validity measure adjusted rand index (ARI) and the internal cluster validation indices of DBI, Dunn index (DI) and SC. The results of the proposed approach are also compared with two state-of-the-art MST-based clustering algorithms, i.e., B-MST [13] and information theoretic MST-based (ITM) clustering [14], where the results suggest better performance of the proposed approach in the majority of the cases.

The rest of the paper is organized as follows: Sect. 2 presents the related work; Sect. 3 explains the proposed solution covering chromosome structure, reproduction operators and the fitness functions. Section 4 lists the detailed experiments, obtained results and discussion, including comparison with the state-of-the-art methods. Finally, Sect. 5 concludes the paper with a few of the future directions.

## 2 Related work

Clustering data and extracting useful patterns attracts interest from an assortment of fields. The MST-based clustering is specifically useful to identify clusters with irregular boundaries [15]. MSTs have been used to group data, in the fields of biology [16], pattern recognition [16], and image processing [17,18]. There are many clustering algorithms available depending on the type of data and the problem statement at hand. Broadly, these can be divided into three categories; partitioning algorithms, hierarchical algorithms, and graph-based algorithms. This section covers the literature that presents clustering using MSTs and the evolutionary algorithms.

Zhong et al. [19] use the *k*-means approach for extracting MST-based clusters from a graph. Their approach utilizes divide-and-conquer scheme to produce an approximate MST having a time complexity of O $(N^{1.5})$, where $N$ is the number of nodes in a complete graph. The approach initially partitions the data into $\sqrt{N}$ clusters using $k$-means algorithm and then these clusters are combined using an exact MST algorithm. The algorithm's performance is evaluated using real-world and synthetic datasets. Internally, Prim's algorithm is employed to extract the MSTs. The computational cost of their framework is dominated by the number of partitions initially created through $k$-means. However, this can be reduced by producing same sized partitions. The approach can also be applied to larger datasets. Perim et al. [13] report an MST-based heuristic called B-MST for clustering. An objective function consisting of tightness and separation index (TSI) is utilized to guide the algorithm. The method is reported to work on co-expression network topology. A local search procedure is developed for TSI minimization. The results are compared with established methods, such as, gene ontology and ARI. The authors report that B-MST produced superior results as compared to other complex clustering algorithms. Muller et al. [14] present an information-theoretic clustering algorithm. Information-theoretic grouping produces non-convex groups by expressing the numerical information in data through the likelihood density function. It is based on Euclidean minimum spanning tree, which is a fast and efficient optimization algorithm. The edge weights in the Euclidean minimum spanning tree represent the Euclidean distances between points. It is a non-parametric algorithm. The only known parameter is the number of classes. Zhou et al. [20] present an adaptive MST-based clustering algorithm (AMST). Their proposal is useful to extract irregular shaped clusters. It works by initially determining the optimal number of partitions in the region using a validity index. Afterwards the candidate clusters are evaluated for their significance. The validity index takes into consideration both compactness and isolation of data. This enables to select the best portion of an MST for clustering. Their approach is compared with static MST (SMST) and dynamic MST (DMST) clustering approaches. The AMST method extracts multiple clusters from the data as opposed to the single cluster formed by SMST and DMST. Simulations also support better accuracy of AMST in comparison with SMST and DMST. Zhou et al. [21] present two MST-based clustering algorithms using Euclidean distance. These include the $k$-constrained algorithm and unconstrained algorithm. The $k$-constrained algorithm extracts an MST from a graph and repeatedly removes edges from it based on a predefined constraint to form $k$ clusters. Whereas, the unconstrained algorithm partitions the data into multiple clusters without describing the number of clusters. This is done by reducing the overall standard deviation of the edges in the Euclidean minimum spanning tree. The algorithm is named maximum standard deviation reduction algorithm (MSDR). It is evaluated using four benchmark datasets from the UCI repository and is compared with the scale-free minimum spanning tree clustering algorithm (SFMST) and $k$-means approach. Although the authors attempt to reduce the time complexity of their approach by using sorting in each node, however, it is still a performance bottleneck. The work in [9] is like the one presented in [21]. Wang et al. [22] introduce an MST-based clustering algorithm that detects clusters by removing inconsistent edges. Edges are considered inconsistent if their weight is larger than the average edge weight of the MST. The algorithm also removes outliers based on density. Density-based outliers are considered for object status, which is the ratio between local density of the object's neighbors and the local densities of their neighboring objects. Density for each object is calculated by assigning an object a local outlier factor (LOF). LOF is compared with a threshold. If LOF is higher than the pre-specified threshold, it is considered an outlier. Several experiments are reported by comparing the solution against other clustering algorithms where the proposal performs better in identifying a relatively small number of density-based outliers during MST construction. Jothi et al. [23] propose two MST-based clustering algorithms. Most of the MST-based clustering algorithms first generate a complete graph from the input dataset and then perform clustering. This usually takes $O(n^2)$ time. The work in [23] aims to reduce the time for constructing MST. The execution time of their approach is $O\left(n^{\frac{3}{2}}\right)$. However, the MST is constructed with computational time of $O\left(n^{\frac{3}{2}}lgn\right)$. Experiments are used to perform evaluations using four artificial datasets. The results show that their algorithm reduces running time as well as gives efficient clustering. Yu et al. [25] explain the hierarchical structures in MST. They state that the already established algorithms and the proposed theories are inadequate for the identification and explanation of the clusters in trees. They devised a method which can first identify a cluster in a tree and subsequently grow into MSTs. The tree agglomerative hierarchical clustering (TAHC) method is introduced for the identification of the clusters in MSTs. The approach is efficient for the detection of the clusters in artificial trees. Normalized mutual information is utilized to quantify the similarity between the underlying real clusters and the clusters detected by the TAHC method. Their method is reported to have application in the identification of the clusters. Xu et al. [16] present a framework to represent multi-dimensional gene expression data as MST. The framework avoids the loss of information during the partitioning process. Their MST-based clustering approach utilizes three objective functions, namely, clustering through removing long MST edges, iterative clustering, and globally optimal clustering. The first objective function is to minimize the total edge distance of multiple trees by partitioning an MST in $k$ subtrees. The second objective function optimizes the $k$-clustering by reducing the total distance between cluster centers and their data points. The third objective function partitions the MST into $k$ trees by globally minimizing the distance between the data points and the cluster centers. This is done by grouping the data points around the best representative. The framework is tested using three datasets. Huang et al. [26] present a modified density-based minimum spanning tree (MST) clustering algorithm. Their algorithm has two phases; a density-based micro-cluster and MST-based tree formation. The initial clustering is performed using the density-based part that detects noise. In the next step the $k - 1$ longest edges of the MST are removed, which results in the generation of $k$ clustering. The clustering results are optimized using an objective function. They further report that micro-MST-cluster algorithm performs better and exhibit good clustering effect. Their algorithm is reported to have various applications in domains with large datasets. Zhong et al. [27] present a hierarchical clustering method by splitting and merging an MST. They employ MST to guide the processes of merging and splitting. During the splitting process the higher degree MST vertices are selected as initial prototypes, while $k$-means is used to split the dataset. The merging process involves the filtration of the subgroup pairs. The neighboring pairs are considered for merging.

As evident from the above literature survey and to the best of our knowledge, the majority of work in MST-based clustering focuses on partitioning the MST for better cluster formations. The use of an evolutionary approach is missing that may produce promising results. This proposal focuses on using evolutionary algorithm for optimizing the MST-based clusters. The works closely related to this proposal are B-MST clustering [13] and Information Theoretic MST-based (ITM) clustering [14]. These are used for comparing the results of the current proposal. The key difference between the proposed work and previous approaches is that it is not based on any assumption about the underlying data. Like in case of ITM, absolute continuous data distribution is assumed. Additionally, in the previous works complex pointer-based data structures were utilized for storing the MSTs and the extracted clusters. This increased the processing time. However, in this case, the trees are avoided to permanently store clusters; instead, one-dimensional arrays are utilized. There is a limited utility of trees while extracting the MSTs, however its cost is minimized by keeping less population size.

## 3 Proposed solution

This proposal presents an MST-based clustering approach to extract optimized clustering using (1+1)-ES. Previously, work has been reported that indicate MST-based clustering as an efficient method for clustering because of its ability to extract arbitrary shaped clusters and outliers [9]. This work initially extracts multiple MSTs from a graph and later ES is used to optimize the clusters represented by these. The (1+1)-ES has benefits in solving optimization problems due to its simplicity and flexibility of strong response in varying circumstances [28]. The advantage of ES over genetic programming (GP) is its ability to avoid premature convergence thus resulting in better clustering. A key issue in ES is its population size. Small population causes ES to converge too quickly, however larger population waste computational resources. The ES population here consists of ten individuals. The initial individuals of the ES population are MSTs extracted from the input graph using Prim's algorithm by selecting varying nodes as the root. Depending on the input graph there may be multiple MSTs. However, this number is usually small thus restricting our framework to have a population size of ten chromosomes. Once the ES population is initialized, one child is produced from each parent through mutation resulting in a parent child pool of 20 chromosomes. Fitness of both, parent, and the child solution is computed. If the child's fitness is higher than the parent, it replaces the parent in the next iteration. Otherwise, the parent chromosome is moved to the next generation. The rate at which mutations occur is often small because large mutation rates alters the structure of the chromosome quickly thereby resulting in the loss of good genetic material in highly fit individuals. The Davies–Bouldin index (DBI) [29] is utilized as a fitness function in the proposed solution. Since a lower value of DBI indicates better clustering, the individuals with minimum DBI values survive in the next generation. This proposal requires the datasets to be represented as a graph by considering each data point as a node and edge indicating distance between them. The proposal is evaluated using various distance measures for the edge weight, such as: Euclidean, Minkowski, Chebyshev, Correlation, Mahalanobis and city block. Figure 1 shows the overall working of the proposed solution. For the proposed system, the input dataset is represented as a graph. The data samples become the graph nodes and an edge between two nodes represent their Euclidean distance. For $N$ samples, we have an $N \times N$ adjacency matrix. Where, each cell containing a non-zero entry shows the distance between two nodes, which is computed by applying the distance measure to its attributes.

### 3.1 Chromosome encoding

To initialize the ES population multiple MSTs are derived from a single graph by arbitrarily selecting a node as the MST root. This results in dissimilar MSTs. The MSTs are derived using Prim's algorithm. However, to have multiple MSTs in the graph there must be multiple edges having the same weight. This result in multiple MSTs having same total weight, but varying shape. Figure 2 shows two MSTs

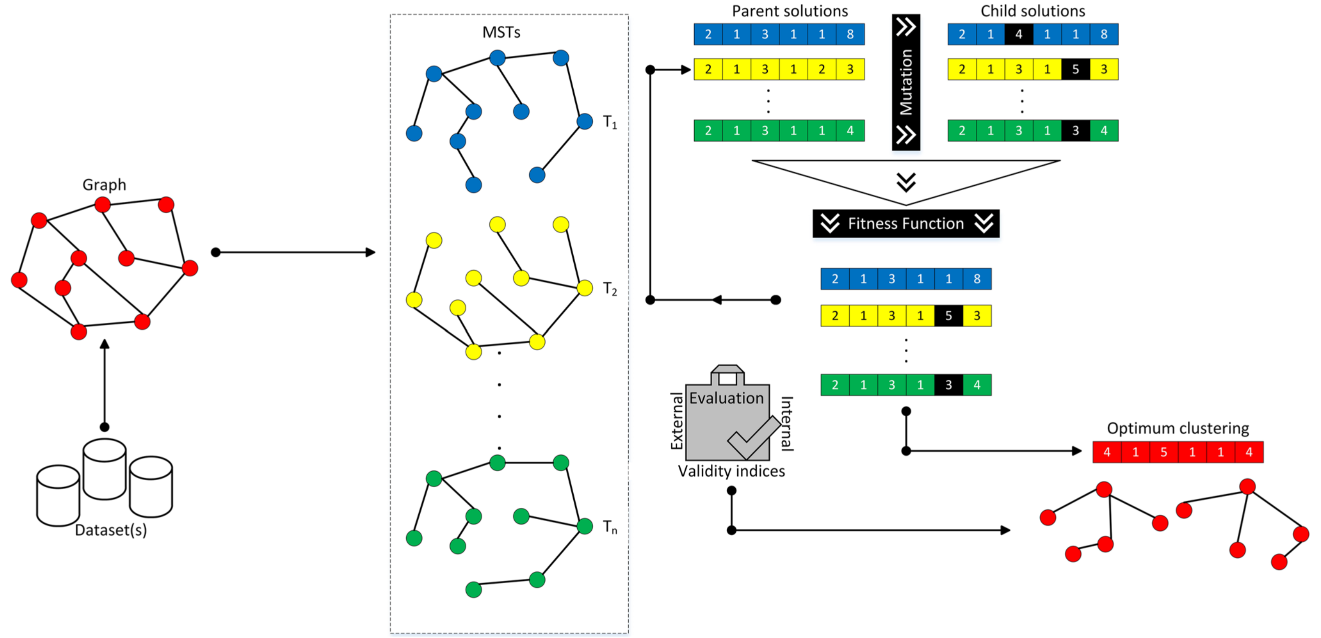

**Fig. 1** Overall system working

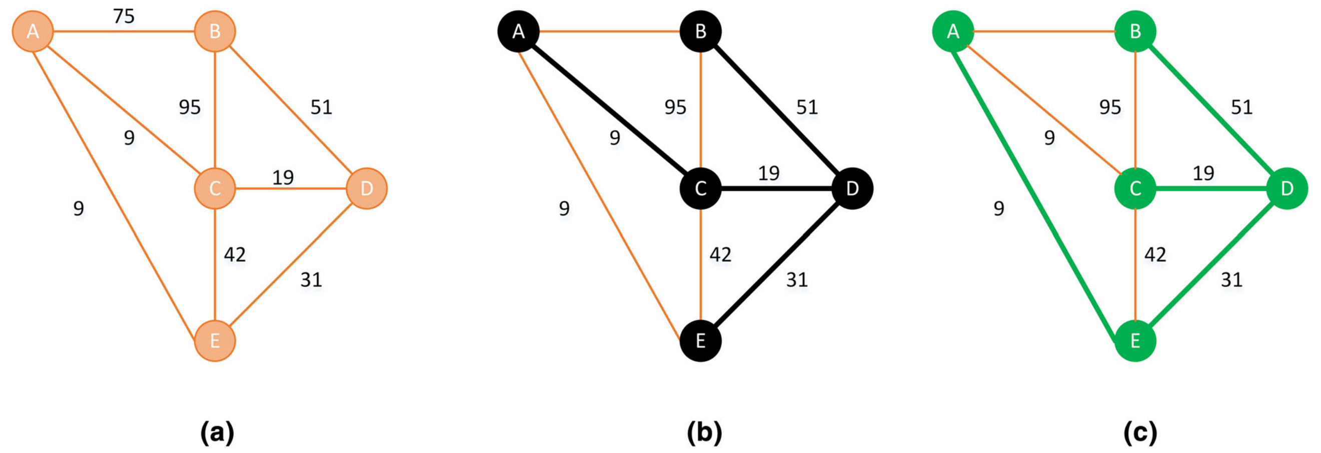

**Fig. 2** A sample graph with two of its MSTs **a** input graph, **b** MST-1, **c** MST-2

extracted from the sample graph. Once multiple MSTs are extracted from the graph, the next step is to extract coherent partitions (i.e., clusters) from these MSTs. Initial clusters from each MST are generated using a specific threshold. For a given $MST^1$, edges are extracted from the MST and added to cluster $C^1_{MST}$, this process is repeated until the cumulative weight of the cluster $C^1_{MST}$ exceeds a threshold $(t_c)$. This threshold is the ratio between the total weight of an MST divided by the number of clusters. When the sum of the cluster's edges reaches a value greater than the threshold, the cluster growth is stopped [See Eq. (1)].

$$\text{Threshold } (t_c) = \left( \sum_{k=0}^{n} W_k \right) / n_c \qquad (1)$$

Where, $n_c$ is the total number of clusters and $w$ is the sum of all the edges in the MST. Upon exceeding the threshold, cluster $C^2_{MST}$ is constructed and the process is repeated until all edges in the MST are exhausted. This approach is used in all MSTs for creating clusters. From $M$ MSTs, we have $M$ sets of clustering formations. The chromosomes are represented as a one-dimensional array where a cell containing $-1$ is used to separate clusters. These cells are called separators. The consecutive series of cells in the chromosome (until a cell having $-1$ is reached) represents one cluster. Each cell can contain an integer between 1 and $n$, where $n$ is the total number of nodes in the graph. This integer value represents the node number of the graph. Once a node is assigned to a cell in the chromosome, it cannot repeat in other cells.

**Fig. 3** Pseudocode to generate clusters using MSTs

```
Procedure-1: Generate N MSTs using prim's algorithm

Input:      Datasets D;
Output:     N MSTs of the graph from D;

Read the dataset $D$ and represent as a graph $G$
Store $G$ in an adjacency matrix $AM_G$
Set $N$ as the number of MSTs where
        $1 \le N \le$ total number of nodes (G)
Set $i$=0
Repeat While ($i<N$)
        Select vertex $v$ as a root where
                $1 \le v \le N$
        Call Prim's algorithm, Prims($AM_G$,V)
```

```
Procedure-2: Generate M clusters from the N MSTs

Input:          N MSTs;
Output:         M clusters;

Set the threshold, $t_c$ (See Eq. (1))
Set $i$=0
Repeat While ($i<N$)
        Select MST $N_i$ for clustering
        Repeat While($Cluster_{weight}<t_c$)
                Add edges from $M_i$ to cluster $C_{MST}^j$
        Store cluster $C_{MST}^j$
```

The initial assignment of the nodes to the chromosome's cell comes from the extracted MSTs. This provides the ES with a logical initial clustering formation to start with. Figure 3 lists the pseudocode that generates clusters using the MSTs.

### 3.2 Objective function

The Davies Bouldin Index (DBI) is used as a fitness to evaluate the cluster quality represented by the ES chromosomes. DBI is a clustering algorithm evaluation metric which checks for the inter-cluster as well as intra-cluster similarity. Being an internal evaluation scheme, DBI validates the clustering based on quantities and dimensions inherent to the dataset. Equation (2) shows the mathematical formulation of DBI, where $n$ represents the number of clusters, $\sigma_x$ represents the average distance of all the nodes in the cluster to the center of the cluster $c_{i,}$ and $d(c_i, c_j)$ is the distance between the centroids of two clusters $c_i$ and $c_j$.

$$DBI = \frac{1}{n}\sum_{i=1}^{n} \max_{i\neq j}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right). \qquad (2)$$

The minimum value of DBI indicates better clustering. The time complexity of computing DBI is $O(d(K^2+N))$.

### 3.3 Reproduction

Mutation is the only reproduction operator used in ES to guide the population towards convergence. Population's each individual is mutated to produce one offspring. The mutation rate of 5% is set in this proposal. For mutation, two sets of nodes are utilized. The first set includes the original cluster's nodes (called the cluster-nodes) and the second set is of nodes that are used for mutation, referred to as the mutation-nodes. The mutation-nodes and the cluster-nodes are two disjoint sets. Nodes in set cluster-node and the ones in mutation-nodes have equal chances to be selected for the replacement and mutation respectively. For example, consider a dataset with nine nodes numbered as 1, 2, 3, 4, 5, 6, 7, 8, and 9. We have a cluster-node set with members 2, 3, 5, 7 and mutation-nodes set having 1, 4, 6, 8, 9 as members. If the mutation rate is 50%, it means that two nodes are randomly selected from the mutation-nodes to be replaced by any two randomly selected nodes from the set cluster-nodes. This results in a new cluster with changed structure. However, the connectivity of various nodes is validated to be present in the original graph before the mutated chromosome is accepted. This mechanism avoids creation of any false-negative clusters. Figure 4 demonstrates an example mutation.

## 4 Experiments and results

This section lists the experiments and their results. The section shows the outcomes of experiments for the proposed approach referred to as evolution strategy towards clustering (ES-TCL) from this point onwards. Performance of ES-TCL

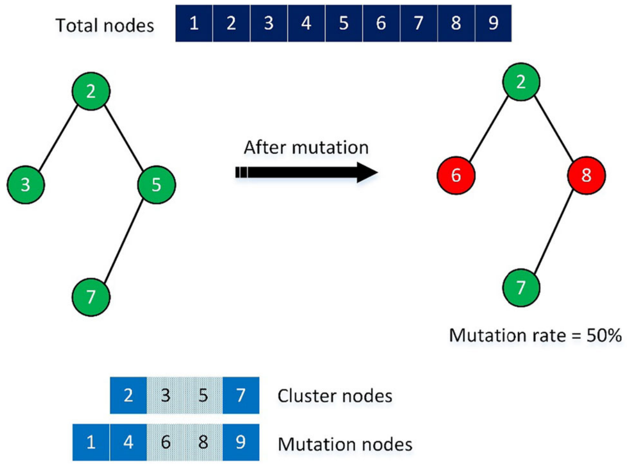

**Fig. 4** An example mutation

is compared with B-MST [13] and Information Theoretic MST-based (ITM) clustering [14], these being the closely related work to the current proposal. B-MST is proposed in [13] that use R and the igraph[1] library [30] for implementation. ITM is proposed in [14] and is implemented using python. ES-TCL is evaluated using eleven benchmark datasets. Seven of these are microarray datasets with known clusters, and remaining four are taken from the UCI repository.[2] The comparison is based on both internal and external cluster validity indices. These include, DBI, DI, SC, and ARI. Table 1 lists the description of the eleven datasets. *Microarray* data represent the measurement technology used for gene expression. Gene expression is the process of finding the rules due to which the information is stored in DNA. *BreastA* dataset is about breast cancer produced using one-channel oligonucleotide with 98 objects and 1213 attributes. *BreastA* is originally clustered into three classes with 51, 11, and 36 samples. *BreastB* is also a breast cancer dataset produced using two-channel oligonucleotide with 48 objects and 1213 attributes. *DLBCLA* stands for diffuse large B-cell lymphoma A, it is a dataset having 141 objects and 661 attributes. *CNS* and *LungA* datasets are utilized in [31] and [13] respectively. *Novartis* represent MultiA gene expression dataset having 5565 genes which are normalized for reduction to 100 and 103 objects. The *digits*, *vowels*, *vehicle,* and *iris* datasets are taken from the UCI repository. Table 2 lists the parameter setting for the ES. The ES is run for 1000 iterations having a population size of ten chromosomes. A mutation rate of 5% is used in the experiments, however, convergence is also evaluated on three other mutation rates.

**Table 1** Datasets description

| Dataset | Samples | Features | Classes |
|---|---|---|---|
| BreastA | 799 | 98 | 3 |
| BreastB | 800 | 49 | 4 |
| DLBCLA | 661 | 141 | 3 |
| DLBCLB | 661 | 180 | 4 |
| CNS | 112 | 9 | 4 |
| LungA | 197 | 188 | 4 |
| Novartis | 799 | 20 | 4 |
| Digits | 1797 | 64 | 10 |
| Vowel | 990 | 10 | 11 |
| Vehicle | 846 | 18 | 4 |
| Iris | 150 | 4 | 3 |

**Table 2** EA parameter settings

| Parameter | Value(s) |
|---|---|
| Number of populations | 1 |
| Initial population size | 10 |
| Reproduction operators | Mutation only |
| Mutation rate | 1, 5, 10, 15, and 25% |
| Stopping criterion | 1000 iterations/convergence |
| ES type | (1+1)-ES |

### 4.1 Cluster validity measures

Once a chromosome represents a clustering formation of the given graph, its quality, i.e., fitness is evaluated using the DBI. However, additional clustering quality indices will help in evaluating the formed clusters by the proposed approach. For this purpose, both internal and external cluster validity indices are utilized. The internal validity index does not require prior knowledge about the clustering structure [32]. Whereas, the external validity indices require this information. External validity indices compare two portions on equality. Other than DBI, the two internal validity indices and one external validity index used for evaluation are listed as follows.

#### *4.1.1 Dunn index (DI)*

Dunn index (DI) is an internal measure for gauging the results of a clustering algorithm. It aims to find solid and disjoint clusters. DI is the ratio between intra-cluster and inter-cluster similarity. Equation (3) lists the formula to find the intra-cluster similarity.

$$intra_{cluster} = \frac{1}{n}\sum_{i=1}^{c}\sum_{a\in Ci}||a - centeri|| \tag{3}$$

[1] http://igraph.org/.

[2] http://archive.ics.uci.edu/ml/datasets.html.

where, $n$ is the number of points in a cluster, $c$ is the total number of clusters, $C_i$ is the cluster and *centeri* is the center of $C_i$. The intra-cluster distance is to be minimized. The inter-cluster similarity shows the separation between clusters which measure the distance between them. For good clustering, it should be maximum. Formula for inter-cluster similarity is given in Eq. (4).

$$inter_{cluster} = \sum\nolimits_{k=0}^{n} \min\left(\|\mathrm{C}i - \mathrm{C}j\|^2\right), \begin{matrix} i = 1, 2, 3, \ldots, c-1 \\ j = 1, 2, 3, \ldots, c \end{matrix} \tag{4}$$

Where, $C$ represents the cluster center.

$$\mathrm{DI} = \frac{\min_{1 \le j \le C}(\text{inter\_cluster})}{\max_{1 \le J \le C}(\text{intra\_cluster})} \tag{5}$$

where, *min (inter-cluster)* represent the minimum distance between two clusters. *Max (intra-cluster)* is the maximum distance between two points in cluster $k$. Dunn index is inversely proportional to $mix(inter - cluster)$. The higher values of DI represent good clustering.

### 4.1.2 Silhouettes coefficient

Silhouettes coefficient (SC) provides a graphical representation of the clustered objects. SC is an internal validity measure showing the similarity and dissimilarity of the object assigned to a cluster. If the object is well clustered, it is more connected to its own cluster's members, and disconnected to other clusters. For example, if an object i is assigned to cluster C1 then average dissimilarity AVGa(i) of i with all other objects of C1 is calculated. Next, the average dissimilarity of i to objects of other clusters except C1 is computed. The SC value ranges between −1 and 1. Formula for computing SC is listed in Eq. (6). The higher SC value is considered better.

$$SC(i) = \frac{AVGb\,(i) - AVGa(i)}{max\{AVGa\,(i)\,,\,AVGab(i)\}} \tag{6}$$

### 4.1.3 Adjusted rand index (ARI)

ARI, an external cluster validity index, finds the resemblance between two clusters by identifying how much two clusters are like each other. Consider the data points in a dataset represented as a set *SD*.

$$SD = \{d_1, d_2, d_3 \ldots d_n\} \tag{7}$$

Let two clusters of *SD* be $C1 = \{a_1, a_2, a_3 \ldots d_s\}$ and $C2 = \{b_1, b_2, b_3 \ldots b_s\}$

**Table 3** Notations used in ARI formula

| Variable | Description |
|---|---|
| a | Represent the number of elements that share the same class in cluster C1 and same class in C |
| b | Represent the number of elements having different classes in cluster C1 and different classes in C2 |
| c | Represent the number of elements that share same class in C1 but assigned different classes in C2 |
| d | Represent the number of elements having different classes in cluster C1 but share same class in C2 |

Where, $C1$ is an external index containing $a_i$ classes, and $C2$ is a result of clustering algorithm containing $b_i$ classes. The formula for ARI is:

$$ARI = \frac{a + b}{a + b + c + d} \tag{8}$$

Table 3 lists the description of $a$, $b$, $c$, and $d$. The value of ARI ranges between 0 and 1. If both clusters are same, ARI is 1. The value $a + b$ shows the connection between $C1$ and $C2$, while $c + d$ shows disparity between two clusters. The higher ARI value indicates better clustering formation.

## 4.2 Distance measures

ES-TCL utilizes Prim's algorithm to extract initial MSTs from a given graph and later clusters are formed. All this involves computation of distance between various nodes. In the experiments six distance measures are utilized, including: Euclidean, Chebyshev, Minkowski, Mahalanobis, correlation, and city block distance. These distance measures are shown in Eq. (9–14).

$$d(a, b) = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + \cdots + (b_n - a_n)^2} \tag{9}$$

$$d_{chebyshev}(\mathrm{a, b}) = \max_i \mid a_i - b_i \mid . \tag{10}$$

$$d_{minkowski}(\mathrm{a}, b) = \left(\sum\nolimits_{i=0}^{n-1} |a_i - b_j|^p\right)^{1/p} \tag{11}$$

$$d_{mahalanobis}(\mathrm{a}, b) = \left(\left(\vec{a} - \vec{b}\right)^T S^{-1}\left(\vec{a} - \vec{b}\right)\right)^{1/2} \tag{12}$$

$$d_{correlation} = \frac{n\left(\sum_{i=0}^{n-1} a_i b_i\right) - \left(\sum_{i=0}^{n-1} a_i\right)\left(\sum_{i=0}^{n-1} b_i\right)}{\sqrt{\left[n\sum_{i=0}^{n-1} a_i^2 - \left(\sum_{i=0}^{n-1} a_i\right)^2\right]\left[\sum_{i=0}^{n-1} b_i^2 - \left(\sum_{i=0}^{n-1} b_i\right)^2\right]}} \tag{13}$$

$$d_{cityblock} = \sum\nolimits_{i=1}^{N} \mid a_i - b_i \mid \tag{14}$$

### 4.3 ES results

The ES-TCL is executed over the eleven datasets for 1000 iterations. After each iteration, fitness of the best individual (from the parent child pool) is recorded. This enables to compute the population's average fitness indicating the convergence or otherwise of the proposed methodology. Figure 5 shows the convergence using five datasets, i.e., *BreastA*, *BreastB*, *CNS*, *DLBCLA*, and *DLBCLB*. Depending on the features and other dataset specific characteristics the fitness function values, i.e., DBI may vary across the datasets. Due to this reason, these values are normalized in Fig. 5 for demonstration purpose. All the datasets converge before 1000 iterations. Figure 6 shows the convergence speed on four mutation rates. These results represent the average values obtained from ten runs. Each run evolved the population for 1000 iterations. The figure indicates quicker convergence of 5% mutation, thus the same is opted for further experiments.

### 4.4 Comparison

For the purpose of comparison, B-MST and Information Theoretic MST-based (ITM) clustering are utilized, these being the closely related recent approach to ES-TCL. Both, the proposed algorithm and B-MST, are distance-based clustering algorithms. For the similarity between objects, six distance measures were evaluated (Sect. 4.2). ES-TCL obtained the results using these six measures (one at a time), and a comparison was made with B-MST. Out of the eleven datasets utilized in this study, seven are microarray datasets, which were originally used in the evaluation of B-MST. Table 4 shows the comparison of the proposed solution with B-MST using Euclidean distance. ES-TCL and B-MST were

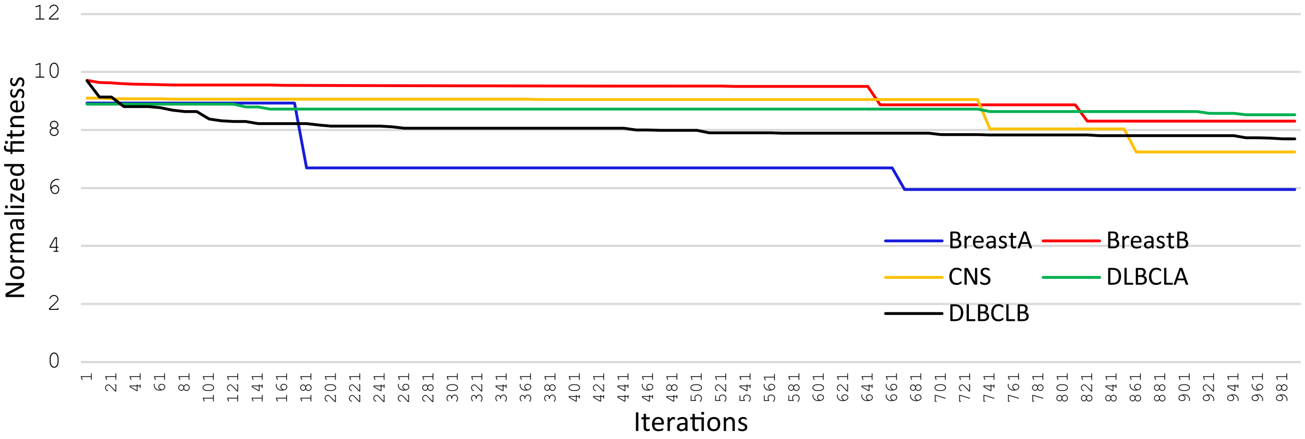

**Fig. 5** (1+1)-ES convergence graphs for five datasets

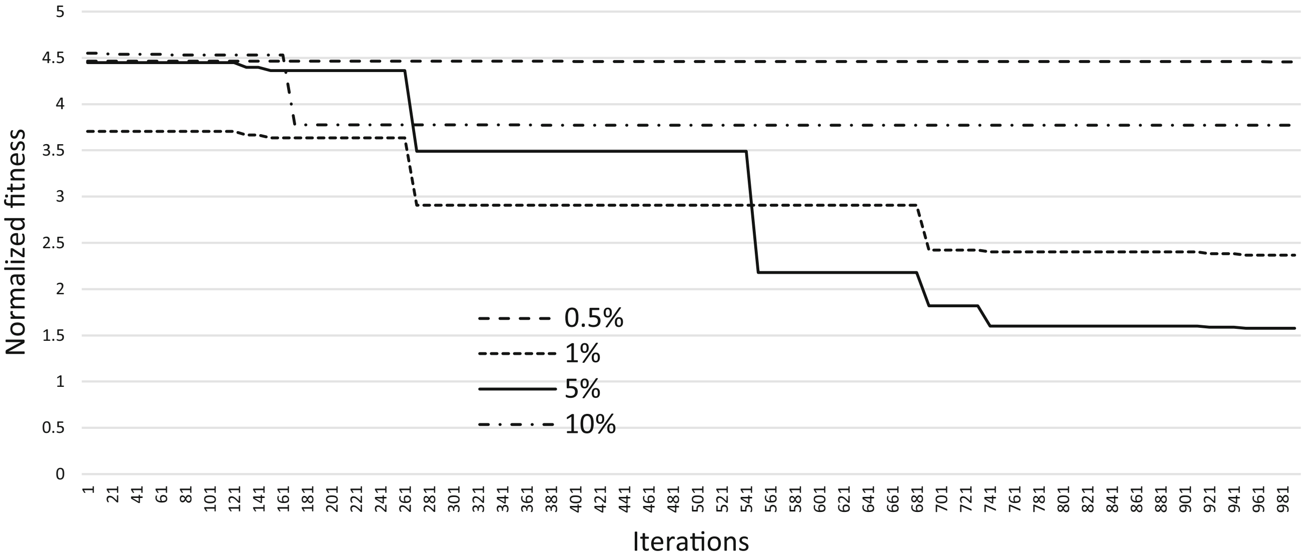

**Fig. 6** Four mutation rates and convergence speed

**Table 4** Cluster validity indices using Euclidean distance

| Datasets | ES-TCL | | | | | B-MST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DBI | DI | SC | ARI | Time (s) | DBI | DI | SC | ARI | Time (s) |
| BreastA | **2.666415** | 0.05752 | **0.143474** | −0.00798 | **148.5644** | 5.1695 | **0.0801** | −0.01558 | **−0.002** | 789.0522 |
| BreastB | **2.427889** | 0.05636 | **0.270697** | **0.00164** | **170.8247** | 4.5508 | **0.0872** | −0.01321 | −0.064723 | 1178.732 |
| DLBCLA | **4.077493** | 0.00362 | **0** | **0.0287** | **88.67897** | 38.8197 | **0.0091** | −0.71704 | −0.2190631 | 858.0941 |
| DLBCLB | **4.042141** | 0.0886 | **0.289725** | −0.0078 | **115.4174** | 5.3449 | **0.0916** | −0.30726 | **−0.003** | 1327.828 |
| CNS | **3.609224** | 0.01106 | −0.50689 | **0.0045** | **11.26453** | 5.3449 | **0.0916** | **−0.30726** | −0.003 | 1327.828 |
| LungA | **1.945492** | **0.0708** | **0.599852** | −0.164 | **18.50534** | 59.1341 | 0.0045 | −0.81183 | **−0.0259** | 55.16616 |
| Novartis | **2.8132099** | 0.034 | **0.163783** | **0.0106** | **143.4517** | 13.4414 | **0.1394** | −0.41247 | −0.0064 | 1337.777 |

Bold values indicate the proposed approach performs better

executed five times and the average value was used for evaluation. The lowest value for DBI and the highest values for DI, SC, and ARI indicate better clustering. The table presents the validity indices of partitions generated by ES-TCL and B-MST. The results suggest that using the Euclidean measure, the proposed approach has better DBI values for all datasets as compared to the B-MST, while DI value is better only for *LungA* dataset. The SC value for ES-TCL is better for all datasets except *CNS*. Similarly, ES-TCL gives better ARI values for *BreastB*, *DLBCLA*, *DLBCLB*, *CNS,* and *Novartis*. Figure 7 shows an SC plot using the *LungA* dataset. The dissimilarity between objects was also computed using Chebyshev distance for ES-TCL and B-MST. The DBI, DI, SC, ARI values for B-MST and ES-TCL are listed in Table 5. The minimum value of DBI is achieved for ES-TCL on all datasets. ES-TCL has better DI values for four among the seven datasets. ES-TCL has better SC value for all the datasets with an exception of *CNS*. ARI values for three datasets, i.e., *BreastA*, *DLBCLA*, and *DLBCLB*, are better for the ES-TCL. The proposed approach takes minimum running time on all datasets except for *CNS* as compared to B-MST. Table 6 lists the validity indices using Minkovski distance. As is the case for Euclidean and Chebyshev distances, ES-TCL has better DBI values for all datasets. It has better DI values only for two datasets; *CNS* and *lungA*. SC values of ES-TCL are better for five datasets. Table 7 mentions the DBI, DI, SC, and ARI measured when correlation is used for clustering in ES-TCL and B-MST. Four datasets have better DBI values using correlation with the current proposal. For DI, no dataset gives better results for ES-TCL. Four datasets have better SC values by using correlation. For ARI, ES-TCL has better values for all datasets, except *lungA* and *Novartis*. Table 8 shows the results using Mahalanobis distance. For this distance measure, all the datasets produce better DBI values excluding *CNS*, over the proposed solution. DI values of ES-TCL are worst only for *BreastA* and *CNS* datasets. Excluding the *CNS* dataset, SI value on all datasets are better using the proposed solution. *Novartis* is the only dataset that does not provide good values on the proposed solution. Table 9 lists the results using city block distance. The results show that DBI is better for all datasets, DI is better for four datasets, i.e., *DLBCLA*, *DLBCLB*, *CNS*, and *Novartis*.

Analysis of the results reveals that the ES-TCL performs better than B-MST in most of the cases. However, there are few instances in the results where B-MST's performance is better. Since the objective function used by ES-TCL is the DBI, it has a success rate of 85.71% for this metric. Considering time, other than the six instances, ES-TCL consumed less running time than B-MST. This is due to the fewer individuals in the ES-TCL's population as compared to B-MST. Additionally, the initial individuals of the ES-TCL were a good estimate of the cluster formation in shape of MSTs. This enabled the approach to converge quickly.

Performance of ES-TCL is also compared with ITM. For this comparison, four UCI repository datasets were used, including: *digits*, *vowel*, *vehicle*, and *iris*. These datasets were also evaluated using three internal and one external validity index. The comparison is also performed on running time. Table 10 lists the results. The current proposal produces better DBI for all the datasets. The four datasets, *digits*, *vowel*, *vehicle*, and *iris* also produce better DI for ES-TCL. ITM is a Euclidean distance based clustering algorithm. While comparing it with the ES-TCL, the highest value among all distance measure is selected. However, it is observed that the ITM performed better using the cluster validity indices of SC and ARI. A deeper investigation of the results in Table 10 reveals that on average the clustering solution provided by ES-TCL is four times better than the one provided by ITM based on the DBI metric. Whereas, the clustering solution provided by ITM is only half or one-third times better than the ones provided by ES-TCL. The ES-TCL is slower than ITM due to being an evolutionary approach.

### 4.5 Discussion

The proposal presented in this work aims at optimizing the clusters represented by the MSTs using evolutionary computation approach. For this purpose, (1+1)-ES is utilized.

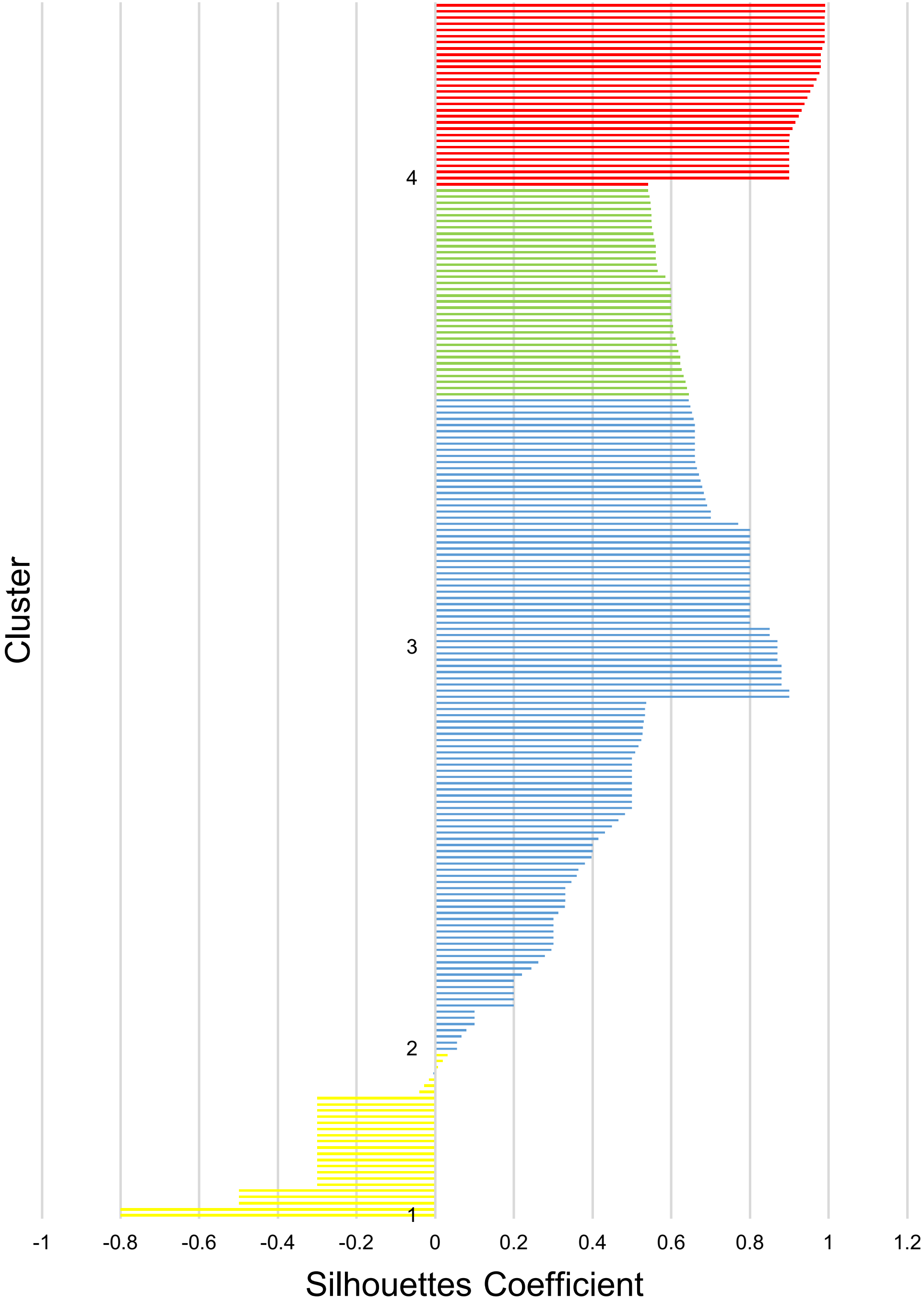

**Fig. 7** Silhouettes coefficient plot using the *LungA* dataset

There are many other evolutionary approaches, including: genetic algorithms, genetic programming, evolutionary programming, and differential evolution. However, the (1+1)-ES seems to be the most suitable evolutionary approach for the problem at hand. The reason for this is the less complications in the reproduction procedure due to the mutation-only strategy as compared to the other algorithms in this domain. Additionally, the problem at hand starts with an initial well-calculated solution instead of a random initial population. This makes ES more suitable for the current task. The ES population is restricted to only ten individuals because of the limited number of unique MSTs extracted from the same input graph. Increasing the population size would not only cause duplication, but will also increase the running time of ES-TCL. DBI, a widely-used cluster validity index guides the ES-TCL. DBI has the advantage of being an internal cluster validity index, that quantifies the quality of clustering using features inherent to the dataset. However, once the ES-TCL converges, the final clustering formations are evaluated using two other internal and one external cluster validity

**Table 5** Cluster validity indices using Chebyshev distance

| Datasets | ES-TCL | | | | | B-MST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DBI | DI | SC | ARI | Time (s) | DBI | DI | SC | ARI | Time (s) |
| BreastA | **2.970359** | 0.0729 | **0.19363** | **0.1624** | **168.5211** | 7.4286 | **0.0902** | 0.14498 | 0.07033 | 678.8684 |
| BreastB | **2.649981** | 0.01842 | **0.286087** | −1319 | **137.645** | 4.7746 | **0.2124** | −0.2138 | **−0.0089** | 1158.405 |
| DLBCLA | **3.793666** | **0.0476** | **0.038762** | **0.0677** | **90.78201** | 16.5012 | 0.0141 | −0.6101 | −0.0085 | 624.1618 |
| DLBCLB | **3.626372** | **0.1245** | **0.484702** | **−0.006** | **124.8141** | 4.5276 | 0.0675 | −0.1504 | −0.1556 | 1118.444 |
| CNS | **5.219949** | 0.01115 | −0.49281 | −0.08496 | 12.54441 | 12.9055 | **0.0159** | **−0.1454** | **0.0145** | **8.511814** |
| LungA | **1.921503** | **0.0926** | **0.275887** | −0.1788 | **17.82112** | 5.7497 | 0.006 | −0.2989 | **−0.0096** | 32.42853 |
| Novartis | **2.529076** | 0.0432 | **−0.00736** | −0.0135 | **133.0246** | 35.8225 | **0.0796** | −0.6665 | **0.0012** | 988.1966 |

Bold values indicate the proposed approach performs better

**Table 6** Cluster validity indices using Minkovski distance

| Datasets | ES-TCL | | | | | B-MST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DBI | DI | SC | ARI | Time (s) | DBI | DI | SC | ARI | Time (s) |
| BreastA | **2.670957** | 0.06484 | **0.153419** | −0.00713 | **152.4506** | 5.1695 | **0.0801** | −0.01558 | **−0.002** | 789.0522 |
| BreastB | **2.433051** | 0.0594 | **0.266573** | **0.01348** | **162.72** | 4.5508 | **0.0872** | −0.01321 | −0.0013 | 1268.31 |
| DLBCLA | **4.164795** | 0.00348 | **−0.56559** | **0.04004** | **90.94402** | 38.8197 | **0.0091** | −0.71704 | −0.0044 | 679.461 |
| DLBCLB | **4.19728** | 0.00322 | −0.5219 | **0.01714** | **94.87888** | 5.3449 | **0.0916** | **−0.30726** | −0.003 | 1118.084 |
| CNS | **3.605427** | **0.0111** | −0.41779 | **0.00553** | 11.45757 | 6.0591 | 0.0097 | **−0.18641** | −0.0235 | **8.912353** |
| LungA | **1.945492** | **0.0708** | **0.599825** | −0.164 | **18.60181** | 59.1341 | 0.0045 | −0.81183 | **−0.0259** | 35.12061 |
| Novartis | **2.8132099** | 0.034 | **0.163783** | **0.0106** | **136.6229** | 13.4414 | 0.1394 | −0.41247 | −0.0064 | 1046.138 |

Bold values indicate the proposed approach performs better

**Table 7** Cluster validity indices using correlation

| Datasets | ES-TCL | | | | | B-MST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DBI | DI | SC | ARI | Time (s) | DBI | DI | SC | ARI | Time (s) |
| BreastA | **2.670957** | 0.06484 | **0.153419** | **−0.00713414** | **152.4506** | 5.4377 | **0.0727** | −0.13726 | −0.28378 | 1342.194 |
| BreastB | **2.433051** | 0.059398 | **0.266573** | **0.013475** | **162.72** | 2.6754 | **0.1483** | −0.18848 | −0.0086 | 1362.543 |
| DLBCLA | 4.164795 | 0.00348 | −0.56559 | **0.04004** | **90.94402** | **2.1164** | **0.1084** | **0.453231** | 0.0105 | 746.7688 |
| DLBCLB | **4.19728** | 0.00322 | −0.5219 | **0.01714** | **94.87888** | 9.5643 | **0.1138** | **−0.28557** | −0.0118 | 1148.729 |
| CNS | 3.605427 | 0.01108 | −0.41779 | **0.005531742** | 11.45757 | **2.5324** | **0.0285** | **−0.35062** | −0.0063 | **8.035242** |
| LungA | **1.945492** | 0.0708 | **0.599825** | −0.164 | **18.60181** | 7.7934 | **0.0916** | −0.78754 | **−0.0299** | 31.10301 |
| Novartis | 2.813209912 | 0.034 | **0.163783** | 0.0106 | **136.6229** | **2.6751** | **1.178** | −0.62493 | **0.16044** | 987.8204 |

Bold values indicate the proposed approach performs better

indices, namely, DI, SC, and ARI. This enables to evaluate the results over other independent metrics previously unknown to the proposed algorithm. The ARI has an added advantage of computing the clustering accuracy even in the absence of the class labels [33]. The proposal is compared with two other MST-based clustering approach, namely: B-MST and ITM. ITM utilizes MSTs, whereas, B-MST employs an evolutionary computing approach for clustering in addition to the utility of MSTs. This makes the comparison rational.

The current proposal is compared with B-MST using six distance measures based on four cluster validity indices and time. The results are listed in Tables 4, 5, 6, 7, 8, and 9. Using the seven datasets, five performance measures (four validity indices and time), and six distance measures a total of 210 indicators are presented for B-MST and ES-TCL. Where, the ES-TCL perform better in 143 instances turning to be 68.0952% better than B-MST. Overall, ES-TCL performs best with the Euclidean distance having an average accuracy of 74.29% for all datasets and all cluster validity indices. Considering all datasets and all validity indices, ES-TCL performs worst using Mahalanobis and city block distances. For the DBI, ES-TCL performs best, i.e., 85.71% of the times, it achieves better performance than B-MST considering all distance measures. ES-TCL performs better

**Table 8** Cluster validity indices using Mahalanobis distance

| Datasets | ES-TCL | | | | | B-MST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DBI | DI | SC | ARI | Time (s) | DBI | DI | SC | ARI | Time (s) |
| BreastA | **1.951134** | 0.1495 | **0.234792** | **0.00888** | **107.7252** | 5.4429 | **0.2318** | −0.18131 | −0.3037 | 893.8644 |
| BreastB | 25.66471 | **0.18904** | **0.29525** | **0.4684** | **118.3054** | **4.1185** | 0.1442 | −0.05782 | −0.138466 | 1250.203 |
| DLBCLA | **1.602811** | **0.5443** | **0.240131** | −0.16781 | **59.66713** | 8.3338 | 0.0705 | −0.38419 | **0.006** | 726.1468 |
| DLBCLB | **1.664109** | **0** | **0.443355** | **0.00466** | **46.59076** | 7.1445 | **0.2734** | −0.0408 | 0.0029 | 1244.952 |
| CNS | 4.2972 | 0.01736 | −0.28381 | **−0.02852** | 19.31994 | **3.4248** | **0.0731** | **0.522244** | −0.0287 | **7.48139** |
| LungA | **1.992492** | 0.0608 | **0.679725** | −0.154 | **20.10122** | 6.8834 | **0.0987** | −0.77721 | **−0.0289** | 32.11214 |
| Novartis | **4.928132** | **0.702** | **−0.14529** | −0.0322 | 162.3264 | 11.7694 | 0.0023 | −0.52068 | **0.0253** | **38.244** |

Bold values indicate the proposed approach performs better

**Table 9** Cluster validity indices using city block distance

| Datasets | ES-TCL | | | | | B-MST | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DBI | DI | SC | ARI | Time (s) | DBI | DI | SC | ARI | Time (s) |
| BreastA | **2.734299** | 0.06556 | **0.163923** | **−0.02094** | **143.3229** | 11.3856 | **0.0829** | −0.23492 | −0.0031 | 766.7153 |
| BreastB | **2.393452** | 0.03522 | **0.234028** | −0.15882 | **174.8822** | 5.0094 | **0.0838** | −0.55813 | **−0.0082** | 1312.105 |
| DLBCLA | **3.196549** | **0.0053** | **−0.37437** | **0.1515** | **92.37199** | 13.0132 | 0.004 | −0.61012 | −0.10678 | 715.1776 |
| DLBCLB | **3.325546** | **0.13902** | 0.379567 | −0.01522 | **117.5778** | 7.6459 | 0.0969 | **0.39193** | **0.05604** | 1245.631 |
| CNS | **3.421404** | 0.0059 | −0.46296 | **−0.0354** | 15.46642 | 5.2355 | **0.0276** | **0.357144** | −0.0368 | **7.294779** |
| LungA | **2.483765** | **0.0698** | **0.67835** | −0.1581 | **20.36352** | 11.7694 | 0.0023 | −0.52068 | **0.0253** | 38.244 |
| Novartis | **3.239041** | **0.2855** | **0.184355** | **0.0184** | **133.1547** | 42.4816 | 0.0896 | −0.71526 | −0.0025 | 1155.687 |

Bold values indicate the proposed approach performs better

**Table 10** Cluster validity indices for ES-TCL and ITM

| Datasets | ES-TCL | | | | | ITM | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DBI | DI | SC | ARI | Time (s) | DBI | DI | SC | ARI | Time (s) |
| Digits | **2.486632** | **0.09828** | −0.16032 | 0.16698 | 1960.86 | 4.0147 | 0.0937 | **0.117218** | **0.838** | **70.45772** |
| Vowel | **2.188279** | **0.0421** | −0.14139 | 0.0107 | 144.6044 | 11.3628 | 0.0411 | **0.053141** | **0.195** | **0.63281** |
| Vehicle | **2.002779** | **0.0599** | −0.38933 | −0.0121 | 136.2124 | 4.0349 | 0.0123 | **0.071473** | **0.141** | **0.461641** |
| Iris | **2.868464** | **0.0306** | −0.39544 | 0.07024 | 14.75993 | 9.5988 | 0.0197 | **−0.00159** | **0.882** | **0.165507** |

Bold values indicate the proposed approach performs better

than B-MST 71.43% of the times considering SC. However, it performs only 23.81% times better than B-MST for DI. Considering time, B-MST consumes less running time and executes quicker than B-MST 88.10% of the times for all datasets and distances. Reason for this is ES-TCL's initial population being computed logically and the lesser number of individuals in each iteration. Keeping in view the above discussion, ES-TCL works at its optimal to optimize the clusters with the DBI as a fitness function and Euclidean/Chebyshev/Minkovski distance as a measure to compute node detachment. With this configuration, ES-TCL performs better than B-MST for all datasets (Tables 4, 5, 6).

ITM is the other MST-based clustering approach used to compare the performance of ES-TCL. ITM's approximate optimization formulation leads it to be an efficient algorithm with low runtime complexity. Whereas, ES-TCL being an evolutionary computing-based approach requires additional time for convergence. The results in Table 10 show that ES-TCL performs better than ITM on all datasets using the cluster validity indices of DBI and DI. ITM does perform better on the cluster validity indices of SC and ARI. Investigating this in depth reveals that the solution produced by ES-TCL is at least two times better based on DBI and DI in comparison to the ones produced by ITM. Whereas, the better clustering solutions provided by ITM based on SC and ARI are only at maximum, half times better than those provided by ES-TCL. However, the key strength of ITM remains to be its much less computation time. Overall, ES-TCL performs better than ITM and B-MST using all cluster validity indices, all datasets, and all distances. Thus, generalizing the results.

Like any other research, there are some limitations of the proposed work. Although this work has performed a detailed set of experiments on evaluating the ES-TCL's performance, where this proposal generally performs better than the two closely related clustering approaches. However, a limitation of the work is duplication of MST's. Though care has been taken for the datasets considered in this study to discard any duplicate MST to be considered as a candidate solution, however, for datasets with many duplicate MSTs ES-TCL will suffer with performance issues. The proposed framework, ES-TCL, can be utilized for various software visualization tasks that benefit from clustering [34,35] and in extracting social circles in the ego networks [36,37].

## 5 Conclusions and future work

This work proposed a MST-based clustering procedure to extract coherent groups from a dataset represented as a graph. An input dataset was transformed into a graph where, nodes of the graph represented the samples and an edge indicated the distance between them. Multiple MSTs from a graph were extracted using Prim's algorithm. The (1+1)-ES was utilized for the optimization of the MST-based extracted clusters. The ES used DBI as its guiding function. Eleven benchmark datasets were used to evaluate the performance of the proposed methodology. The (1+1)-ES-based cluster optimization approach, named, ES-TCL, was executed using 10 chromosomes for 1000 iterations. A mutation rate of 5% was used. The results were compared with two state-of-the-art MST-based clustering algorithms, B-MST and ITM. For this comparison three internal validity indices (DBI, DI, and SC), and one external validity index, ARI was used. The proposed solution was also compared with the two algorithms with respect to execution time. B-MST and the proposed solution both use distance-based clustering, so these were evaluated using seven microarray datasets. The results suggested that the proposed solution on average performed better as compared to B-MST and ITM. The proposed approach can be explored further in the future. A limitation of the proposed solution is that it finds $n$ MST in a graph which may cause to build identical MSTs. This needs to be looked into in the future. Evolutionary approaches are slow by their nature, in the future other faster optimization techniques can be used to extract MST-based clustering formations.

## References

1. Datta, S., Datta, S.: Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics **19**(4), 459–466 (2003)
2. Shen, H., Yang, J., Wang, S., Liu, X.: Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets. Soft Comput. **10**(11), 1061–1073 (2006)
3. Srinivasan, G.: A clustering algorithm for machine cell formation in group technology using minimum spanning trees. Int. J. Prod. Res. **32**(9), 2149–2158 (1994)
4. Thawonmas, R., Ashida, T.: Evolution strategy for optimizing parameters in Ms Pac-Man controller ICE Pambush 3. In: IEEE Symposium on Computational Intelligence and Games, pp. 235–240 (2010)
5. Eberhart, R.C., Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: IEEE Evolutionary Computation, pp. 94–100 (2001)
6. Wu, F., Mueller, L.A., Crouzillat, D., Pétiard, V., Tanksley, S.D.: Combining bioinformatics and phylogenetics to identify large sets of single-copy orthologous genes (COSII) for comparative, evolutionary and systematic studies: a test case in the euasterid plant clade. Genetics **174**(3), 1407–1420 (2006)
7. Huang, A.: Similarity measures for text document clustering. In: Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, pp. 49–56 (2008)
8. Zha, H., He, X., Ding, C., Simon, H., Gu, M.: Bipartite graph partitioning and data clustering. In: Proceedings of the tenth international conference on Information and knowledge management, pp. 25–32 (2001)
9. Grygorash, O., Zhou, Y., Jorgensen, Z.: Minimum spanning tree based clustering algorithms. In: Tools with Artificial Intelligence, pp. 73–81 (2006)
10. Halim, Z., Kalsoom, R., Baig, A.R.: Profiling drivers based on driver dependent vehicle driving features. Appl. Intell. **44**(3), 645–664 (2016)
11. Hussain, S.F., Mushtaq, M., Halim, Z.: Multi-view document clustering via ensemble methods. J. Intell. Inf. Syst. **43**(1), 81–99 (2014)
12. Abraham, A., Guo, H., Liu, H.: Swarm intelligence: foundations, perspectives and applications. In: Swarm Intelligent Systems, pp. 3–25 (2006)
13. Pirim, H., Ekşioğlu, B., Perkins, A.D.: Clustering high throughput biological data with B-MST, a minimum spanning tree based heuristic. Comput. Biol. Med. **62**, 94–102 (2015)
14. Müller, A.C., Nowozin, S., Lampert, C.H.: Information theoretic clustering using minimum spanning trees. In: Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium, pp. 205–215 (2012)
15. Zahn, C.T.: Graph theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. **C–20**(1), 68–86 (1971)
16. Xu, Y., Olman, V., Xu, D.: Clustering gene expression data using a graph-theriotic approach: an application of minimum spanning trees. Bioinformatics **18**, 536–545 (2002)
17. Gonzalez, R.C., Wintz, P.: Digital Image Processing. Addison-Wesley, Reading, MA (1987)
18. Xu, Y., Olman, V., Uberbacher, E.C.: A segmentation algorithm for noisy images: design and evaluation. Pattern Recognit. Lett. **19**, 1213–1224 (1998)
19. Zhong, C., Malinen, M., Miao, D., Fränti, P.: A fast minimum spanning tree algorithm based on K-means. Inf. Sci. **295**, 1–17 (2015)
20. Zhou, R., Shu, L., Su, Y.: An adaptive minimum spanning tree test for detecting irregularly-shaped spatial clusters. Comput. Stat. Data Anal. **89**, 134–146 (2015)
21. Zhou, Y., Grygorash, O., Hain, T.F.: Clustering with minimum spanning trees. Int. J. Artif. Intell. Tools **20**(01), 139–177 (2011)
22. Wang, X., Wang, X.L., Chen, C., Wilkes, D.M.: Enhancing minimum spanning tree-based clustering by removing density-based outliers. Digit. Signal Process. **23**(5), 1523–1538 (2013)

23. Jothi, R., Mohanty, S.K., Ojha, A.: Fast minimum spanning tree based clustering algorithms on local neighborhood graph. In: International Workshop on Graph-Based Representations in Pattern Recognition, pp. 292–301 (2015)
24. Tzortzis, G., Likas, A.: The MinMax k-Means clustering algorithm. Pattern Recognit. **47**(7), 2505–2516 (2014)
25. Yu, M., Hillebrand, A., Tewarie, P., Meier, J., van Dijk, B., Van Mieghem, P., Stam, C.J.: Hierarchical clustering in minimum spanning trees. Chaos: an interdisciplinary. J. Nonlinear Sci. **25**(2), 023107 (2015)
26. Huang, G., Dong, S., Ren, J.: A minimum spanning tree clustering algorithm based on density. Adv. Inf. Sci. Serv. Sci. **5**(2), 44 (2013)
27. Zhong, C., Miao, D., Fränti, P.: Minimum spanning tree based split-and-merge: a hierarchical clustering method. Inf. Sci. **181**(16), 3397–3410 (2011)
28. Abraham, A., Nedjah, N., Mourelle, L.: Evolutionary computation: from genetic algorithms to genetic programming. In: Genetic Systems Programming, pp. 1–20 (2006)
29. Halim, Z., Waqas, M., Hussain, S.F.: Clustering large probabilistic graphs using multi-population evolutionary algorithm. Inf. Sci. **317**, 78–95 (2015)
30. Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal Complex Syst. **1695**(5), 1–9 (2006)
31. Bandyopadhyay, S., Mukhopadhyay, A., Maulik, U.: An improved algorithm for clustering gene expression data. Bioinformatics **23**(21), 2859–2865 (2007)
32. Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.: Internal versus external cluster validation indexes. Int. J. Comput. Commun. **5**(1), 27–34 (2011)
33. Iwata, T., Lloyd, J.R., Ghahramani, Z.: Unsupervised many-to-many object matching for relational data. IEEE Trans. Pattern Anal. Mach. Intell. **38**(3), 607–617 (2016)
34. Halim, Z., Muhammad, T.: Quantifying and optimizing visualization: an evolutionary computing-based approach. Inf. Sci. **385**, 284–313 (2017)
35. Muhammad, T., Halim, Z.: Employing artificial neural networks for constructing metadata-based model to automatically select an appropriate data visualization technique. Appl. Soft Comput. **49**, 365–384 (2016)
36. Leskovec, J., Mcauley, J.J.: Learning to discover social circles in ego networks. Adv. Neural Inf. Process. Syst. **25**, 539–547 (2012)
37. Mcauley, J.J., Leskovec, J.: Discovering social circles in ego networks. ACM Trans. Knowl. Discov. Data **8**(1), 4 (2014)

**Zahid Halim** received the B.S. degree in computer science from the University of Peshawar, Pakistan, in 2004, M.S. degree in computer science from the National University of Computer and Emerging Sciences, Pakistan, in 2007, and the Ph.D. degree in Computer Science from the National University of Computer and Emerging Sciences, Pakistan, in 2010. He was with the National University of Computer and Emerging Sciences, Islamabad, Pakistan, as a Faculty Member (Lecturer and then Assistant Professor) from 2007 to 2010. Currently he is an Associate Professor with Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan. His current research interests include machine learning and data mining, probabilistic/uncertain data mining, and human factors in computing.

**Uzma** received the Master of Computer Science degree from the AWKU in 2013 and the M.S. degree in computer system engineering from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan in 2016. She is currently working towards her PhD from the Faculty of Computer Science and Engineering, GIK Institute, Pakistan. She is a member of the Women Engineers Society. Uzma's research interest includes data mining, machine learning, and bioinformatics.