Wireless Pers Commun (2017) 94:2009–2030
DOI 10.1007/s11277-016-3350-5

Evolutionary Clustering Algorithm Using Criterion-Knowledge-Ranking for Multi-objective Optimization

M. Anusha[1] · J. G. R. Sathiaseelan[1]

Published online: 3 May 2016

Abstract There are variety of methods available to solve multi-objective optimization problems, very few utilizes criterion linkage between data objects in the searching phase, to improve final result. This article proposes an evolutionary clustering algorithm for multi-objective optimization. This paper aims to identify more relevant features based on criterion knowledge from the given data sets and also adopts neighborhood learning to improve the diversity and efficacy of the algorithm. This research is an extension of the previous work named neighborhood learning using k-means genetic algorithm (FS-NLMOGA) for multi-objective optimization which maximizes the compactness of the cluster and accuracy of the solution through constrained feature selection. The proposed objective finds the closest feature subset from the selected features of the data sets that also minimizes the cost while maintains the quality of the solution. The resultant cluster were analyzed and validated using cluster validity indexes. The proposed algorithm is tested with several UCI real-life data sets. The experimental results substantiates that the algorithm is efficient and robust .

Keywords Data mining · Neighborhood learning · Multi-objective optimization · Clustering · Feature selection · Genetic algorithm · Criterion learning · Knowledge acquisition · Cluster ranking

1 Introduction

Now-a-days, the researchers are directing their focus to solve multi-objective optimization (MOP) problem. Since the solution to the problem is highly signified in the fields of engineering [1], computer science [2], information technology [3] and so on. Most of the

✉ M. Anusha
anusha260505@gmail.com

1 Department of Computer Science, Bishop Heber College, Trichy, Tamilnadu, India

multi-objective optimization problems have multiple objectives that conflicts each other. Hence, the objectives has to be simultaneously optimized [4]. There are three types are MOPs namely, weighted function, lexicographical approach and Pareto approach [5]. The weighted function is the process of transformation of multi-objective problem into single-objective problem by using certain weighted measure. Lexicographical approach uses ranking methods whereas Pareto approaches consists of several non-dominated solutions [6]. In-order-to solve MOP, there is a need set of optimal solutions, called Pareto-front solutions. These solutions are obtained using Genetic Algorithms (GA) which are subjected to the shape or continuity of the Pareto-fronts [7]. Therefore Pareto-front gained more attention in the field of data mining. Pareto-fronts could be obtained through genetic algorithm. In-order-to mine MOP, there is a need to formulate certain rules such as defining fitness function, adopting accurate genetic operators and so on [8].

Genetic algorithms are heuristic search and optimization technique which provides near optimal solution for an optimization problem. In GA strings are represented as parameters and the collection of parameters depicts population. The objective function associated to the string outlines the degree of goodness of the population [9]. According to Darwin's theory of "Survival of Fitness", the strings are randomly selected then induced to cross-over and mutation [10]. The genetic operations are carried over to produce a new sub-population or off-spring. The process of selecting objects from the population and progressing with genetic operations continuous the selected constraint reached the termination condition or the chosen objective attains the fixed number of generation. GA would enrich to classify or cluster the labeled or unlabeled data set.

Clustering is a most predominant domain in Data Mining where interesting information and hidden data would be revealed in several fields such as Computer Science, Bio-technology, and Petro-chemical [11]. Data could be clustered based on cluster compactness and connectedness. Compactness of the cluster is measured by the overall deviation between the objects and their corresponding cluster centers. Compactness can be stated as:

$$comp(s) = \sum_{c_k \in s} \sum_{i \in c_k} d(i, \mu_k) \tag{1}$$

where s represents number of s, μ_k is centroid of the cluster and $d(i, \mu_k)$ is the distance between ith object and the cluster center. Connectedness is evaluated with the help of closest data points of the cluster centers. It is identified as:

$$conn(s) = \sum_{i=1}^{N} \left(\sum_{j=1}^{L} x_i, nn_i(j) \right) \tag{2}$$

where nn_i is the jth nearest neighbor of the i and L determines number of neighbors that are connected to the one another. There are many clustering techniques namely, k-means algorithm [12], branch-bound technique [13], self-organizing map [14] and so on are available to solve, most of the clustering problems by implementing optimization techniques. In this view, when there is an increase in search space of the population that in turn maximizes the size of the data concurrently. Redundant data or data that is farthest from the cluster center can be truncated by appropriate feature selection for the data set [15].

Feature selection is an eminent pre-processing technique which aims to improve the efficiency of the efficacy of the clustering [16]. Most of the scholars can adopt their preferred feature selection methods in their research work. Since the issues are normally dealt with real-life data sets, it is essential to make proper filter or selection with the help of the approaches namely, wrapper methods, filter approach and hybrid method [17, 18]. Filter method selects feature based on the theoretical properties of the particular data sets. Wrapper method nominate the feature based on learning methodology. Hybrid method is the combination of wrapper methods and filter approaches. Investigators makes use of this selection procedure to acquire highly significant and relevant data for their study. The feature of the specific data set could be evaluated by extracting the sub-string of the original string. The process of evaluation is iterated until it reaches the stopping criteria. Based on the evaluation, sub-population can be generated with certain measure or fitness constraints [19, 20]. In case newly generated sub-population sounds better, then the previous population can be replaced by the new off-spring.

In this paper an interactive evolutionary clustering algorithm (ECMO) is proposed to solve multi-objective optimization problem. The aim of this algorithm is to classify more relevant features in the data set that spectates in maximizing diversity and accuracy with respect to minimize the processing time and cost for clustering the data sets. The rest of the paper is organized as follows. In Sect. 2, a brief introduction to neighborhood learning and feature selection is described. Then Sect. 3, a massive review of some past studied are presented. Thereafter in Sect. 3, proposed ECMO is discussed in detail. Section 5 shows the experimental results obtained from the study. Finally, conclusion and possible research issues are presented in Sect. 5.

2 Background

In this Section, some fundamental concepts of Neighborhood learning is described and certain feature selected to the development of FS-NLMOGA are also discussed.

2.1 Neighborhood Learning and Feature Selection

Precursor NLMOGA, generates the set of objects as population. The objects are selected based on fitness value called knowledge criterion. Neighbor object was selected using crowding distance. Neighborhood learning generates randomly generated objects from the initial population repository. New off-spring could be represented by n-dimensional vector array in the solution space of R^n. Neighborhood learning is performed through superior and inferior data objects of the solution space which substantiate the neighborhood relation between the objects. This method also provides a better cluster compactness for the associated data. The total cumulative learning of the data sets were achieved by comparing the superior with old inferior. Algorithm also triggers the nearer object using the crowding distance that is associated to that particular object. In case, the selected object dominates the fitness value, the superior object could be truncated from the current population. In this way, NLMOGA concludes with superior, inferior and learned objects by retaining the better neighbor and truncating the rest. The extended version of NLMOGA is FS-NLMOGA, where certain theoretical features of the attributes were chosen as selected-key feature for the dataset. This research paper overcomes FS-NLMOGA by adapting criterion-knowledge feature selection method where the features are learnt algorithmically and evaluated using criterion-ranking approach [21].

2.2 Multi-objective Optimization as Clustering

Conventional Genetic Algorithms are single objective which optimizes single criterion in the searching phase. Today, many real-life problems involve in several objectives that are need to be optimized simultaneously. Hence, the problem concerned with multiple objectives poses many solutions, each of which is considered as relatively important and significant. Multi-objective optimization (MOO) problem can be traditionally stated as [22]

$$Vector\ \bar{x} = [x_1, x_2, \ldots, x_n] \tag{3}$$

where $\bar{x}$ are decision variables The $\bar{x}$ could be defined as Pareto optimal solution, when there is no $\overline{xx}$ dominates $\bar{x}$. The Pareto optimal $\bar{x}$ can be defined as

$$\forall i \in 1, 2, 3, \ldots, k = \ > \exists f(\bar{x}) < f((\overline{xx})) \tag{4}$$

Pareto fronts has been considered as subset or closer to the Pareto optimal solution. The equal distribution of solution extends to produce a better trade off. The obtained solution could illustrate the overall performance of the objectives that could project innovation. In this vein, population-based method called genetic algorithm is used handle the situation. Pareto based methods are utilized to archive the population. There are two ways through which population could be archived. They are population based non-Pareto approach and Pareto based non-elitist approach. In population based non-Pareto based approach the data has been processed using vector evaluated genetic algorithm, lexicographic ordering, game theory, gender for identifying objects, contact algorithm and non-generational genetic algorithm [23].

Vector evaluated genetic algorithm uses special operator to analyze the objective function. In lexicographic ordering, optimization of the objectives are devised using ranking given by the user. Game theory allots a player for each objective. Gender for identifying the objectives approach combines several parents to produce a single offspring. Contact algorithm could be designed with the help of fitness function between the relative objects from the Pareto fronts. In case of Non-generated genetic algorithm, multi-objective problem could be converted into a single objective problem to remove noisy data. Pareto-based non-elitist has been categorized as MOGA, NPGA and NSGA. Basically MOGA allocate rank to each individual, based on the ranking schema population would be categorized. All dominated objects are incremented with a specific value than the non-dominated objects. NPGA adopts tournament selection strategy to choose the dominated object. NPGA uses a dummy fitness value that is relevant to the population. Dominated objects are truncated and the population is reshuffled till the population reaches clear clustered data [24].

2.3 Genetic Algorithm in Data Clustering

Data mining is the task of identifying relevant or useful information from the complex data set. At present, many real-world have composite issues, population-based techniques called genetic algorithm is to solve many data mining issues. Since real-life problems have multiple objectives that often conflicts each other, which work synergistically to provide compliance solution with robustness and low cost called soft computing. Genetic algorithms are one of the soft computing techniques which accomplish the challenges through data mining techniques such as clustering, classification, feature selection and so on [23]. Extensive application of genetic algorithm is used for data clustering is achieved with the help point-based method and center-based method. In point-based clustering the length of the data set is predefined with the total number of data objects present in the data set. Nonetheless center-based clustering method the data objects are encoded using cluster centers. Most of the clustering algorithm would not likely to produce optimal solution for high dimensional data set in-order-to overcome this difficulty proper feature selection clustering has to be adopted by using clustering techniques like K-Means, KNN and so on could be hybrid with genetic algorithm to acquire global optimal solution. The performance of the clustering algorithm could be procured by making use of cluster validity measures [24].

2.4 Feature Selection for Clustering

Feature selection is the process of selecting relevant feature or attribute for data clustering. A proper feature selection reduces the dimensional of the subjective space. A better feature selected algorithm can able produce compact clusters. In general, feature selection could be stated as

$$p(\dot{f}) = \min_{f \in \varphi}\{p(F, X)\} \tag{5}$$

where φ is the set of all possible feature subsets and F is the selected feature subset in vector X. P is defined as a key function that could able to decide the quality of the feature with respect to data set [25]. Since, the data labels of the clustering is unknown, filtering approach could restrict the data based on the distribution of values across the vector data sets. In addition to this method, wrapper class uses clustering algorithm to indicate the appropriate feature for the data set. The irrelevant feature can be removed using deviation detection. It is the process of identifying the frequently occurring objects in the data set. Deviation can be detected using statistical approach, distance based approach and deviation based approach. In statistical based approach, the probability model of the dataset is considered to evaluate the deviation. Distance based method calculates deviation using distance measurement between two objects. Deviation based method identifies the irrelevant data based on the properties of the data [25].

3 Literature Survey

Based on the study stated in Sect. 2, an extensive survey on MOGA is discussed in this section. The description emphasis the advantages and the drawback of the algorithms. Lopez-James et al. [26] stated reference point approach which could help to integrate decision-makers preference into MOEA. The algorithm can sort out the location and region of interest during feature searching. It is also noted that size of the relationship is scalable with respect to selected objects and it was caliber with achievement functions and the Pareto fronts.

Gracia-Piquer et al. [27] outlined about prototype-based, label-based and graph-based representations on synthetic data. The algorithm was analyzed based on the accuracy, search space and time cost through clustering algorithms. It was inferred from the algorithm was poor in cluster object convergence. Wang et al. [28] proposed single-objective guided multi-objective optimization to solve multi-objective optimization problem. The data could be learnt from neighbor fields and proceeded with ε-dominance archive. The objective of this work was to minimize the number of clusters with cluster convergence.

However, the algorithm needs improvement in adopting crossover operation based on local search. Wang et al. [29] applied local based application of evolutionary operators to define sub-population using hierarchical clustering. The author described the algorithm with independent local environment-selection and local genetic variation. The algorithm could not produce proper cluster convergence and diversity. Liu et al. [30] introduced decomposition multi-objective optimization evolutionary algorithm called MOEA/D-M2M. It was noticed that computational cost increases, when the value of the cluster along with vector data was maximized. A massive survey on evolutionary clustering for MOPs were depicted in [31, 32].

Li et al. [33] used adaptive operator to determine the performance in optimization process. This methodology was demonstrated with the function named sliding window which was intended to calculate the fitness rate based on rank multi-armed bandit. Carreno Jara [34] handled MOP with the help of *p*-optimality criteria. This criteria adopts inverse crippled solution that can sort out the outstanding performance of objectives. Nonetheless, the algorithm lacks in individual object interaction to procure more precise solutions. Deb et al. [35] suggested a reference point based evolutionary algorithm. Population that close to the reference points were classified into non-dominated levels on the hyper-plane. It was observed that the approach could not adapt any additional reference points that would increase Pareto optimum solutions. Deb and Himanshu [36] extended NASA III to solve many constraint problem. The author illustrated the algorithm with modified elitist operator selection for producing offspring population. Peng et al. [37] recommend multi-objective k-means genetic algorithm which could deliver Pareto front solution set, by not considering the weight of the objects. This approach was evaluated using various cluster validity measures with suitable voting techniques. Detection of outliers has been found as main drawback of this method. Mukhopadhyay et al. [38] defined an interactive genetic algorithm based multi-objective approach that could simultaneously found clustering solution by evaluating the validity measures. The algorithm reduces fatigue of the decision maker by generating only important solutions from the current population.

Xia et al. [39] depicted novel soft subspace clustering for MOP to provide best cluster solution. The approach optimizes two objectives. Firstly, to reduce the difficulty in identifying the subspace of the cluster. An indicator called projection similarity index was used to select the best solution and identification best cluster number is done by combing projection similarity index with gap statistic method. Wang et al. [40] improved the clustering process by introducing a redundant cluster operator which could dynamically modify number of clusters during evolution. Chhavi et al. [41] postulated dynamic recommender system based on evolutionary clustering which improves the prediction accuracy also alleviating the scalability of the temporal dimensions. Nevertheless, computational efficiency has to be improved. Singh et al. [42] has developed Pareto corner search evolutionary algorithm which fetches corners of the Pareto front to identify relevant objectives. It is remarked that there was an inadequate use of selection operator that in turn fails in accuracy. Kirkland et al. [43] focused on centroid based multi-objective clustering that lacks to reduce the number of clusters. Bandyopadhyay et al. [44] introduced priority based ε-dominance to rank the objectives according to their priority. The method calculates ε-values before ranking the objectives. The ranking could be possible based on the priority given by the decision maker. The algorithm can be improved with better ε-coefficient.

Liu et al. [45] investigates on constraint handling techniques which uses relaxation approach for MOPs. This method appends one or more objectives associated with that constraints. Moritz et al. [46] suggested two methods for ranking of MOPs. This ranking methods were used to prune large data-sets of solution to small subset of good solution. Li et al. [47] presented a dual population paradigm that uses two separate co-evolving populations to deal with convergence and diversity simultaneously. The algorithm undergoes certain issues namely, selection of sub-region and crossover operator. Inkaya et al. [48] extracts the density, connectivity and proximity relations among the data points in an adaptive manner. The neighborhood data could be achieved through local characteristics of the data points. However, the algorithm unable to reduce computational complexity while constructing neighbor. Kimovski et al. [49] proposed parallel feature selection for multi-objective optimization approaches to cope with high dimensional data. Parallel feature can be selected with the help of forward–backward algorithm on clustering method to distinguish between genuine and non-genuine data. Nevertheless, algorithm needs improvement in data-sets dimensionality, cluster overlap.

Byers et al. [50] derived solution for unfavorable interaction between the objects. Relationship between the objects were calculated using crowding distance. This method incorporated object function granularity to determine the effects of searching process. It is observed that the algorithm could be improved using ε-dominance to handle unintended interaction. Luque et al. [51] presented a novel parallelization approach using preference relation based on clustering technique to deal with MOPs. The clustering methods helps to find different region using preference based MOEA. Saha et al. [52] devised a framework to identify most relevant set of features for MOPs. The algorithm can able to recognize set of features with ample number of clusters. The compactness of the cluster was classified using point symmetry function. Long [53] stated a constraint handling technique for MOPs. The author considered the constraints like closeness, diversity and feasibility were scaled using elitism metric, cell-based diversity metric and feasibility metric. The author extended the work by altering the optimal sequence method to evaluate the fitness and cell-based density metric [54]. Chen et al. [55] presented a local search framework for MOPs. This method also includes farthest-candidate approach to maintain the cluster diversity. The main weakness could be the trap of local optimal solution on certain test problems. The archiving strategy is replaced with the traditional non-dominated sorting and crowding distance measure proposed in [56]. A filter-based multi-objective feature selection algorithm denoted as FMOFSA, wherein the entropy measure proposed in [57]. The algorithm with a classical filter method called ReliefF [58]. An entropy weighting k-means algorithm for subspace clustering of high dimensional sparse data is introduced in [59]. These observations motivate us to propose a novel ECMO framework. The proposed framework is discussed in the following section.

4 Proposed Algorithm

This section address the issues specified in Sect. 3 by proposing an evolutionary clustering algorithm for MOPs. Primarily, ECMO generates uniform set of objects as the population. Then, the population is treated with three main procedures until the termination condition is satisfied. These three major operations are criterion learning algorithm (CLA), knowledge acquisition algorithm (KAA) and optimal cluster-ranking algorithm (RA). The ultimate goal of CLA is to perform global search based on the discovered criteria and then the knowledge is acquired through constant learning to dominance.

Algorithm 1 ECMO Framework

1. Initialize the Global Population repository GPR.
2. **while** stopping criterion is not met **do**
3. S ← Seclect_Population(GPR)
4. **for** obj ← 1 to *obj* **do**
5. Optimize S^{obj} by criterion learning until f_{obj} cannot be Improved.
6. **end for**
7. **for** each new_generated_population in GP **do**
8. GPR← update_population using Knowledge and rank (new_generated_population)
9. **end for**
10. **end while**
11. Return GPR as result

While RA refine the process by grouping most relevant data with the help of ranking strategy. Figure 1 illustrates the framework of ECMO.

4.1 ECMO Framework

ECMO, concentrates mainly on three functions namely, selecting the solution from the population repository, then by adopting selected objectives extraction of information is learned through the criteria. Based on criteria, neighborhood knowledge could be discovered for the data objects and produce S sub-population. Finally, the archive could be updated using most relevant objects of *obj* using appropriate ranking. A well-structured population repository is maintained to keep Pareto fronts obtained during search. The pseudo-code of ECMO framework is presented in Algorithm 1.

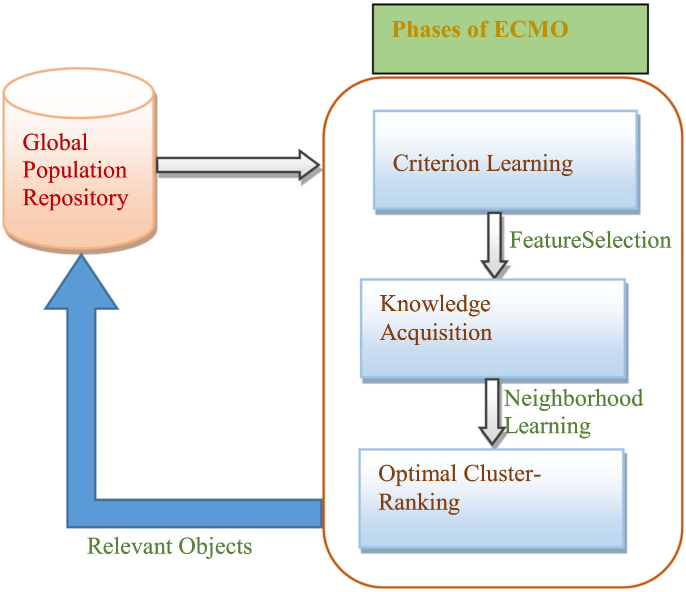


Fig. 1 General framework of ECMO

A general framework of ECMO is designed in Fig. 1.

Various steps of ECMO are demonstrated in the forthcoming sections.

4.2 Criterion Learning Algorithm

In criterion learning, the fitness value of the given objective is learned. Based on the criterion each objective is computed. In this phase, original population is directly taken for feature selection. In a population, the compactness of the individual objects represents the degree of crowding of the area where the individual is located. Hence, the criterion learning (CL) for the particular objective could be scalarized. It is stated as

$$Min\ CL(x) = Max_{K=1}^{OBJ} \alpha^{*}(obj(x) - c_i) \tag{6}$$

The Eq. (6) can able to find even poor expressing points. A small value of $\alpha = .005$ is suggested to choose the high expression points that are nearer to c_i referencing point. Hence, the fitness of the population is calculated as

$$Criterion(x) = \min_{h=1}^{H} CL\left(x, c^{h}\right) \tag{7}$$

In this sequel, a minimum criterion will be created at high expressing regions to individual reference points (obj) for selected feature set (fs). The total number of reference points H can be considered on the basics of

$$H = \binom{Obj + fs - 1}{fs} \tag{8}$$

Since, the depicted reference point is extensively spread over the population, the obtained solution will naturally be a near Pareto solution. The proposed formula is likely to respond even for the user chosen reference points. The pseudo-code for criterion learning is stated in Algorithm 2.

Algorithm 2 Criterion Learning Algorithm (CLA)

1. Initialize the CL,H,x,N
2. **while** stopping criterion is not met **do**
3. Generate new_popualation based on Equation (6)
4. **foreach** H in OBJ
5. Learn criterion for x using (7)
6. **end for**
7. **for** i= 1 to N
8. [criterion, clust_loc]= min(CL(sleep $_{(i\ in\ H)}$, stand)
9. **end for**
10. **end while**
11. Return [criterion, clust_loc]

The algorithm initializes each individual objects x, CL vector is then calculated to find the expressive points on the population. Criterion is learned from each objects using (7) that could produce minimum sleeping-index for the corresponding reference. Hence, by considering the sleeping-index as cluster-location (clust-loc) degree of crowing of the area is generated. In this juncture, there is a chance for data accumulation on the same cluster and therefore some cluster may be unoccupied with data. This problem is fulfilled in next phase called Knowledge Acquisition, where neighborhood learning is suggested.

4.3 Knowledge Acquisition Algorithm

The knowledge is accomplished using 3 steps. These stages are explained below.

4.3.1 Stage 1: Identification of Closely-Related Neighbors

For the given H, ascend all the points in GPR by NLMOGA. Then, construct the ordered-set for H as OH. Let OH_j be jth member of OH. Call NLMOGA to find the neighbors points. The object that is close to H than OH_h is constructed as closely-related neighbors (CRN). This method is explained using Fig. 2.

4.3.2 Stage 2: Identification of Farthest-Related Neighbors

Based on reference point H in OH, the objects that are farthest from H is not considered as neighbor, even though it express the related criterion value. In-order-to overcome this problem, a proper feature selection based on boundary level is adopted. Setting of boundary level the major complexity of this procedure. The distance between the H and the farthest object (fo) from H is calculated using Euclidean distance and stored in the temporary storage file. Then, by choosing proper common neighbor between fo and H, the farthest object can also be grouped as neighbor. The process of identifying the farthest is illustrated in Fig. 3. The time complexity of this procedure is O $(OBJ*GPR^2)$. Hence, the cluster is formed by identifying the neighbors which is connected through common neighbors.

4.3.3 Stage 3: Identification of Indirect Neighbors

The indirect neighbors (IDN) can be identified using farthest-related neighbor (FRN). Let IDN_i be the jth neighbor object of i. in case, j and IDN_i are close to each other, then they are

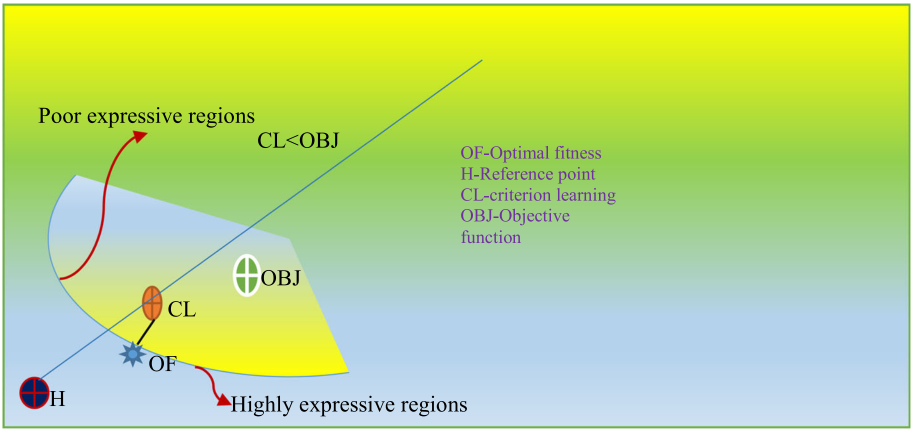


Fig. 2 Criterion learning approach to identify poor and high expressive points

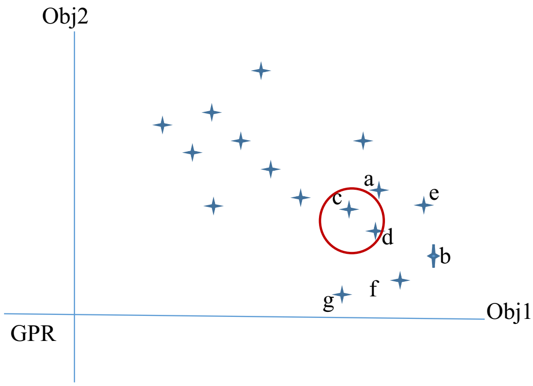


Fig. 3 Example for KA Algorithm. Object d is the closely-related neighbour to c

indirectly neighbor to relative neighbor. Figure 4 demonstrates the method of choosing indirect neighbors. In-order-to balance are indirectly neighbor to relative neighbor. Figure 4 demonstrates the method of choosing indirect neighbors. In-order-to balance the cluster location the objects that are farthest or indirected is shifted close to relative neighbors.

The pseudo-code for this method is outlined in Algorithm 3. Hence, the procedure knowledge acquisition could reduce the formation empty cluster and also preserves diversity.

Algorithm 3 Knowledge Acquisition Algorithm (KAA)

```
Initialize the CL,H,x,N
while stopping criterion is not met  do
    Generate new_popualation from CLA
    foreach H in OBJ
        // search for closely-related neighbor
        nei_sel=NL(x,H)
          if the x<H
            consider x as closet-neighbor of H
          // search for farthest-neighbor
          elseif (x1<x)
            consider x1 as farthest-neighbor of H
          else
              consider x2 as the indirect-neighbor and linked by agent
            end if
    for i= 1 to N
        [criterion, clust_loc]= min(CL(sleep (i in H) , stand)
    end for
end while
Return [criterion, clust_loc,nei_sel ]
```

The cluster balance is maintained using optimal-ranking. The algorithm initializes H along with CL from Algorithm 2. KAA procures the new_population from CLA. Select the reference point in new_population. Apply Neighborhood-learning (NL) on the selected object (x) to the selected reference point. In case the x lies closer to the H, the object x is considered as closest-neighbor of H. In order to accomplish diversity, the farthest object which possess the similar quality of H is considered through the closest-neighbor. The indirect-neighbor is identified through the sub-agent of farthest-neighbor. After the identification of exact knowledge about neighbor, proper cluster is identified. Since, the correct location of the clusterhas been, the procedure reduces the formation of empty clusters (Fig. 5).

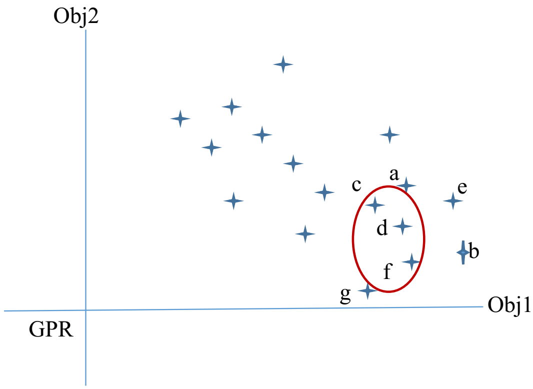


Fig. 4 Example for KA Algorithm. Object f is the closely-related neighbour to d also farther to c. f is farthest-related to c

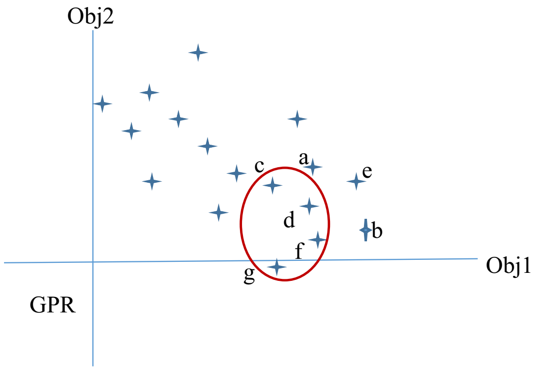


Fig. 5 Example for KA Algorithm. Object g is the closely-related neighbour to f which is farther to c. g is indirectly-related to c

4.4 Optimal Cluster-Ranking Algorithm

Instead of comparing two solutions directly, it is more promising to compare them by its performance with respect to the reference set solutions. There are several methods available for ranking namely, dominance ranking, dominance depth ranking and many more. In this section, a new method of cluster-ranking is proposed. They are two ways through which ranking can be processed, they are best knowledge ranking method and balancing Pareto frontiers. The procedure this method is stated in Algorithm 4. The most important of issue for unconstraint MOP is to protect the diversity and convergence of the solution simultaneously which also assures reliability. Most of the researchers considers the preservation of diversity and compactness as the objective for MOP. These two objectives do conflict each other always. A good diversity may not have better convergence and vice versa. Therefore, optimal cluster ranking address this issue through the proposed algorithm.

4.4.1 Procedure 1: Best Knowledge Ranking

Owing to good objects selection from the group, there is need to associate the performance of each individual objects. The performance of the objects are evaluated in phase II. Pareto fronts are ranked based on the best knowledge generated by the objects. The objects with final common neighbors are connected. Each and every objects in the current population repository holds the separate objective function. Subsequently, the identified final neighbors are grouped as cluster which depends on ranking. The optimal knowledge (OK) for an object is given as

$$OK(x_{ij}) = \sum_{H \in GPR} HOx_{ij} \tag{9}$$

where H is the reference point of the object for the common neighbor O with respect to optimal fitness for the object x. The best knowledge ranking for the selected cluster is identified using

$$best\_pt_{\beta i} = \max_{i \in GPR} (best\_pt)_{OK(x_{ij})} \tag{10}$$

where β_i is the solution of the optimal fitness called best_pt the solutions obtained for this Eq. (10) is known as best solutions for the selected cluster. The process for selecting the best solution is specified in Algorithm 4.

Algorithm 4 Best Knowledge Ranking Algorithm (BKRA)

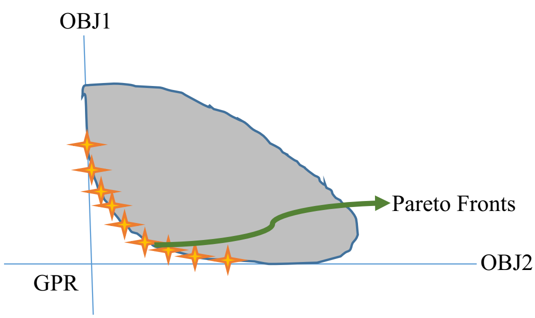
1. Initialize the current population from KAA and criterion from CLA
2. **while** stopping criterion is not met **do**
3. compute the optimal knowledge using Equation (9)
4. **foreach** H in GPR
5. //arrange the soluti0on based optimal knowledge
6 sel_obj=arrange_min_order OK(x_{ij})
7. **end for**
8. best_point for cluster is derived from Equation (10)
9. **end while**
10. Return [sel_obj,best_point]

The algorithm helps to preserve the diversity of the cluster solutions. The selected population is split into number of clusters with the help of best points returned by Eq. (9). Farthest-neighbor is shifted towards the group using the accuracy produced by the objective function of the individual object. Shifted objects has to maintain stabilized cluster for next generation and is discussed in the procedure 2.

4.4.2 Procedure 2: Balancing Pareto Fronts

Stabilizing diversity and convergence is a significant goal of cluster for MOPs. Without proper maintenance, the population may tend to produce only few clusters. Hence, to

Fig. 6 Balacing Pareto fronts with best knowledge ranking



stabilize this problem a small penalty value is transferred to the derived knowledge. A suitable balancing strategy is prosed to uphold the Pareto fronts of the solutions. Balancing of Pareto fronts is explained in Algorithm 5. Diagrammatical representation of balancing Pareto front with the best knowledge ranking for the solution is given Fig. 6.

Algorithm 5 Balancing Pareto Fronts Algorithm (BPFA)

```
Initialize the current ranked population using BKRA, relevancy criteria from CLA
with minimum penalty
while stopping criterion is not met  do
    distribute the population based on best points obtained by Equation (9)
    foreach H in GPR
      x_ij = FRN_shift(x,H)
        sel_obj=arrange_min_order (x_ij )
        next_gen= list (Pareto_fronts)
    end for
          balance=sum(selected feature/number of features)
          fitness_sel_fea=ρ+sum(balance*accuracy)/best-pt
end while
Return [balance, fitness_sel_fea, best_point]
```

Alleviating cluster compactness, divide the population on the basics of best points. The suitable balance for the cluster is persisted by emphasizing KKA to shift the object towards the reference points. Then, by arranging the objects from minimal distance to best point of the cluster. The cluster is balanced using the selected best feature of the cluster among the total number of features of the clusters using Eq. (11). On the whole, the total fitness of the cluster is defined in Eq. (12) by summing up the balance and accuracy with minimum penalty. The accuracy of the cluster is evaluated by classifying possible optimum feature sets against the number of features in the clusters. The formula for depicting accuracy is

$$precison(c) = \sum_{c=1}^{OF} \left(OF_{OK\left(x_{ij}\right)} \right) x_i \tag{11}$$

In order to balance the formation of cluster, distance between the selected features of the chosen clusters searched from the total number of features of the clusters. The Eq. (12) is best preserves the compactness of the cluster.

$$stablize\left(c_{i,j}\right) \sum_{i=1}^{Best\_pt} \frac{c_i(OF1) - \left(c_j OF2\right)}{NF} \tag{12}$$

The best knowledge ranking of the cluster is alleviated through,

$$tot\_clut\_fit = p\left(\frac{precison * stablize}{best\_pt}\right) \tag{13}$$

whereas, p is the small penalty value ≈ 0.05.

5 Experimental Studies

To evaluate the performance and efficacy of the proposed algorithm ECMO, an unsupervised genetic algorithm is discussed in this section.

5.1 Data Set and Experimental Setting

The proposed algorithm is tested ample number of microarray datasets. This does not mean that the algorithm works only with microarray datasets, the framework can work well for general clustering problem. Seven real-life dataset are taken from UCI data repository [60]. Table 1 contains the information about the datasets for the analysis. The algorithm were implemented in 7.6 and executed using Pentium with 2.99 GHZ CPU and 2 GB RAM. The operating system Microsoft Windows XP.

5.2 Parameter Settings

The parameter setting and termination condition are illustrated in Table 2. MOUFSA2 [56] is a feature selection algorithm. This algorithm uses traditional non-dominate sorting with crowding distance instead of archiving strategy also the feature sets adopts negative cardinal rules for the chosen objective.

FMOFSA [57] is a filter-based feature selected multi-objective genetic algorithm where entropy measure is proposed to select the cardinal subset of the objective. ReliefF [58] is a filter based multi-objective genetic algorithm where the neighbors are identified with traditional K-NN. MOEASSC [39] in multi-objective clustering genetic algorithm uses negative weight entropy to maintain the size of the subspace. EMKM [59] uses entropy weighting K-Means and local adaptive clustering (LAC) is used to represent the certainty of dimensions in the identification of the cluster. Proposed ECMO is quite explained in Sect. 4.

5.3 Testing Datasets and Performance Metrics

Seven UCI real-life data sets are used to test the performance and accuracy of the algorithms. The description of the data set are listed in Table 1. The foremost aim of cluster validity indexes is to validate clustering solution. This index is useful in comparing the performance of the cluster. Some widely used cluster validity techniques are the jaccard index, rand index and so on. Suppose T is the true clustering of the dataset and C is a clustering result given by some clustering algorithm. Let $\alpha 1$, $\alpha 2$, $\alpha 2$, $\alpha 4$ denote the number of pair of points belong to the same cluster in T and C and vice versa. The index can able find the points belong to same cluster T not in C and vice versa. We adopted rand index (RI) to compare the performance of the algorithm with the selected real-life datasets. In order to show the efficiency of the proposed algorithm ECMO multi-objective evolutionary clustering algorithm, some real-life datasets are considered. In general, these datasets are

Table 1 Information about datsets

Data sets	Size of the data sets	Number of dimensions	Number of clusters
Wine	178	13	3
Heart	270	13	2
Vechicle	846	18	4
Ionosphere	351	34	2
Secom	1567	590	2
Semeion	1593	256	10

Table 2 Parameter settings for the comparative algorithms

MOUFSA2:	MOEASSC:
Maximum generation: 100	Stopping: MaxGen >= 100
Size of population P: 40	Size of IPop: 100
Size of archive **A**: 3d	Size of EPop: 10
Permitting number of each box t_n: 3	Size of A:200
Weight coefficent **r**: 0.5	Number of totstage: 20
	Value of p_s; 0.5
	Value of p_m: C/D
FMOFSA:	EWKM:
Maximum size of selected features:min[20,d]	y_1: 1,2,5,10,50,100,1000
Maximum generation: (16 {20,d},d–d)/10	Stopping: const $\leq 10^{-9}$
Size of population **P**: 10	
Size of archive **A**: 100	
ReliefF:	ECMO:
The neighbor size is same to that in K-nn	Stopping: MaxGen $\geq$ 200
	Size of the population$_{GPR}$:60
	Crossover probability $_{CP}$: 0.85
	Mutation probability $_{MP}$: 0.18

highly dimensional with many features. Hence, it is essential to find the correct feature for datasets which could help in partitioning the cluster truly. Table 3 shows best performance than comparing the other five algorithms.

ECMO takes 14–17 iterations independently on every dataset for its clustering process. It is noted that the proposed algorithm accomplishes better that other compared algorithms and slightly poor on heart, secom and semion datasets. It is praiseworthy that ECMO prevents the formation of cluster along with good convergence and diversity as shown Fig. 6. It is observed from Fig. 7. ECMO can produce better convergence and diversity for certain datasets. It can be identified from the Table 3 rand index value of the proposed algorithm is comparatively low than other algorithms except few. When the value of RI is maximum, the formation of cluster will be good. Hence, it is certain that ECMO generates better convergence and diversity. Table 4 records the performance of proposed metric best knowledge rank index (BKR) for the real-life datasets. BKR determines the goodness of the cluster with two measures namely, precision and stability which is stated in Sect. 4. The larger the value of BKR implies the proposed algorithm produces minimum diversity

Table 3 Evalution of proposed performance metrix

Data sets	Rand Index	BKR Index
Wine	0.9614	0.9629
Heart	0.66732	0.7216
Vechicle	0.6547	0.6528
Ionosphere	0.7325	0.7329
Secom	0.8759	0.8823
Semeion	0.8903	0.8769

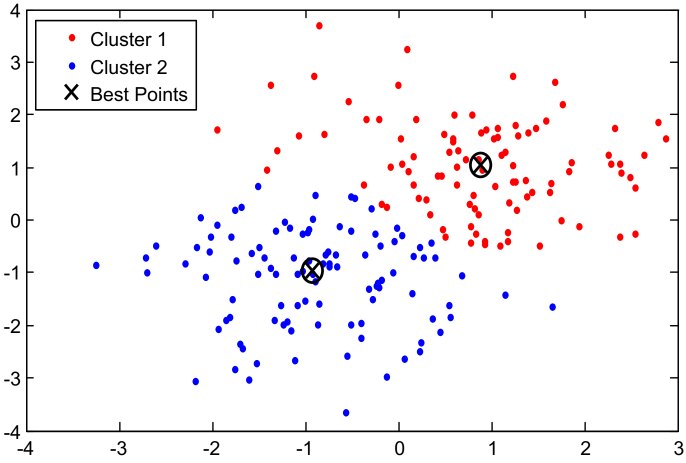


Fig. 7 Best points through CL feature selection

Table 4 Overall accuracy for the comparative algorithms

Data sets	MOUFSA	FMOFSA	RELIEFF	MOEASSC	EWKM	ECMO
Wine	0.7908	0.6642	0.8849	0.9592	0.9337	0.9614
Heart	0.637	0.6328	0.637	0.7139	0.6788	0.66732
Vechicle	0.6449	0.5953	0.6523	0.6533	0.6378	0.6547
Ionosphere	0.7273	0.6979	0.5549	0.6655	0.5889	0.7325
Secom	0.876	0.8756	0.876	0.8752	0.8754	0.8759
Semeion	0.884	0.8577	0.8435	0.848	0.8855	0.8903

and maximum compactness of cluster data. It is witnessed that proposed BKR outperforms traditional RI.

ECMO selects 3 features from the wine dataset. The algorithm can able to find correct number of clusters from this data set. Since, the best feature is identified from CLA along with proper application of KAA, an optimal Pareto fronts are obtained using OCRA. The results are proved in Tables 3 and 4. Hence, we can conclude that ECMO is successful in clustering for its augmented cluster accuracy.

6 Results and Discussion

Experimental results substantiates that the proposed algorithm ECMO, can identify appropriate features set using criterion and produces better clusters by utilizing the procuring the knowledge from the neighbors. The algorithm adopts neighborhood learning from the previous work and the NLMOGA procedure is extended to figure the closest-neighbor, farthest-neighbor and the indirect neighbor. Based on the outcomes of CLA and KAA, an excellent clusters were ranked with more compact and less in diversity. The Tables 3 and 4 reveals that performance of ECMO is better than MOUSFA, FMOFSA, RELIEFF, MOEASSC and EWKM for some datasets. However, ECMO is found to

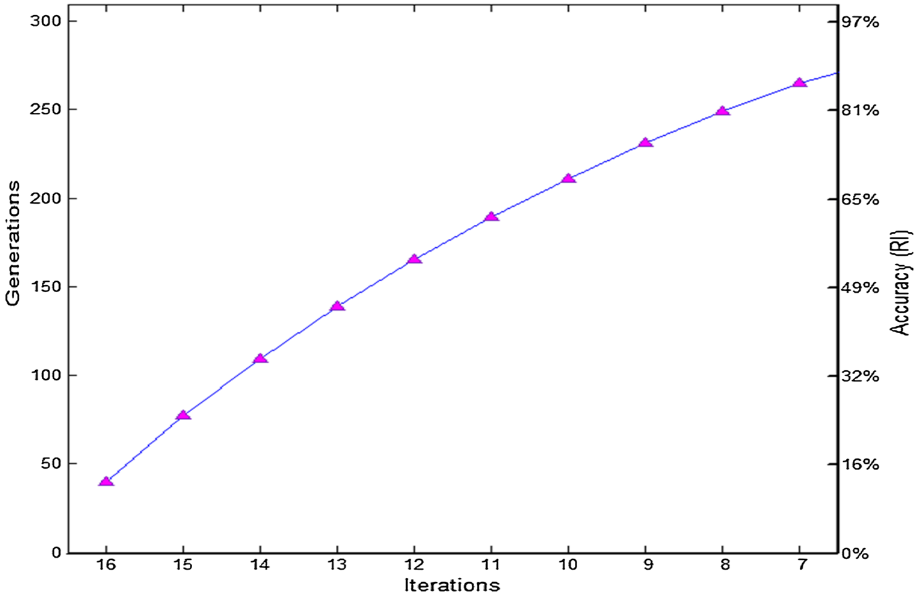


Fig. 8 Pareto fronts of Wine datset

produce maximum BKR index value along with maximum rand index value for almost every best points of the cluster. Therefore, these results indicate that algorithm produce better clusters than the other methods. It is wise notable that same set of objective function will not suit to all the data sets. Figure 8 reports the Pareto fronts for the Wine datasets. The data set wine produces a significant result for the proposed acquisition knowledge algorithm. We record maximum BKR values and maximum rand index values for population generation 200 at 13th iteration.

The CLA helps to find a suitable criterion with best feature selection (OF). OF is used to alleviate the neighborhood learning by generating best point for the feature. The best point of the feature is shown in Fig. 7. Sensitivity of the algorithm can wine and be scaled with population size and the feature selected for the population. The proposed algorithm works well for vehicle datasets. Nonetheless, for the remaining datasets feature increases when the size of the population increases. Even though the algorithm proves the robustness for wine and vehicle data sets, it varies on the selected feature of the datasets. The total computational cost of ECMO consists of three phases. It the summation of three phases along with number of generations. The cost of each phase is evaluated using size of the population (GPR), criterion of the datasets (CL) along with number of the neighbors with respect to the best point of the feature. Hence, the computational cost complexity is O|(GEN*OF)/GPR|. Therefore, the algorithm generates minimum cost for certain data sets.

7 Conclusion

In this paper, a new evolutionary clustering algorithm called ECMO was proposed to address the clustering problem of high-dimensional and low-dimensional data. The approach possess a significant advance for the determination of multi-objective evolutionary problem such as feature selection problem, neighborhood learning and cluster ranking. Moreover, the same criteria has been considered as a promising multi-objective clustering problem too. Although, few proposals have been presented in this area, this article differs in the originality in identifying criterion of the best feature subsequently with suitable knowledge of the neighbor.

The phases of ECMO adopts NASA II for its ground work. The concept criterion knowledge ranking was introduced to attain the two objectives of faster convergence and good diversity of the solutions in MOPs. In this work, we provided a new way of selecting features (CL) from each objectives that intent to depend on the best points of the datasets. The neighbors were identified using KAA. The algorithm can able to recognize more suitable neighbor objects while it possess the similar properties of the selected feature. ECMO shifts the objects position according to their relative proximity. Hence farthest and indirect neighbors can also be the neighbors to the related neighbor (RN) of the current population.

The proposed algorithm generated more relevant clusters which reduces the formation of unfiled clusters. Also, ECMO overcomes the difficulty of determination of weights for each objective for calculating fitness. Usage of rand index demonstrates the best performance of ECMO over the other selected algorithm that were considered for the comparative study. The efficiency outperforms when evaluated using proposed performance metric called BKR index on UCI datasets. However, we do not affirm that ECMO is always better than other algorithms. The weakness of the algorithm has to be studied based on the characteristics of the test problem. There are certain issues that has to be addressed are searching space and the convergence time. In future, we will try to solve these issues. Also, the future direction can be investigation new distance metric and to adopt global searching technique to increase the scalability and robustness of ECMO. A deep study on evolutionary operators on crossover rate and mutation factor is required to maximize the performance of the algorithm in computation cost and time.

References

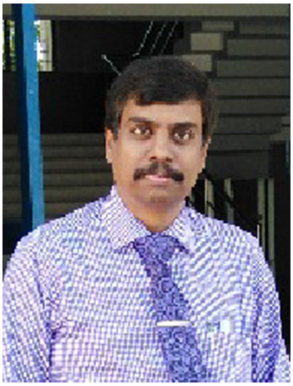
1. Montano, A. A., Cello, C. A. C., & Mezura-Montes, E. (2012). Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Transactions on Evolutionary Computation, 16*(5), 662–694.
2. Das, S., Maity, S., Qu, B.-Y., & Suganthan, P. N. (2011). Real-parameters evolutionary multimodal optimization: A survey of the state-of-the-art. *Swarm and Evolutionary Computation, 1*(2), 71–88.
3. Maulik, U., Bandyopadhyay, S., & Mukhopadhyay, A. (2011). *Multiobjective genetic algorithms for clustering-applications in data mining and bioinformatics*. Berlin: Springer.
4. Cheshmehgaz, H. R., Haron, H., & Sharifi, A. (2013). The review of multiple evolutionary searches and multi-objective evolutionary algorithms. *Artificial Intelligence, 43*(3), 311–343. doi:10.1007/s10462-012-9378-3.
5. Nandaa, S. J., & Pandab, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation, 16*, 1–18.
6. Schutze, O., Laumanns, M., Collo, C. A. C., & Talbi, E. G. (2010). Compting gap free pareto front approximations with stochastic search algorithms. *Evolutionary Computation, 18*(1), 65–96.
7. Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., & Basu, D. K. (2012). A genetic algorithm based region sampling for sampling for selection of local features in handwritten digit recognition applications. *Applied Soft Computing, 12*(5), 1592–1606.
8. Islam, S. M., Das, S., Ghosh, S., Roy, S., & Suganthan, P. N. (2012). An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 12*(2), 282–500.

9. Chatterjee, S., & Mukhopadhyay, A. (2013). Clustering ensemble: A multiobjective genetic algorithm based approach. *Procedia Technology, 10*, 443–449.
10. Kirkland, O., Rayard-Smith, V. J., & de la Iglesia, B. (2011). *A novel multi-objective genetic algorithm for clustering* (pp. 317–326). Berlin: Springer.
11. Mokryani, G., Siano, P., & Piccolo, A. (2013). Optimal allocation of wind turbines in microgrids by using genetic algorithm. *Journal of Ambient Intelligence and Humanized Computing*. doi:10.1007/s12652-012-0163-6.
12. Anusha, M., & Sathiaseelan, J. G. R. (2014). An enhanced K-means genetic algorithm for optimal clustering. In *2014 IEEE ICCIC* (pp. 550–584).
13. Liu, L., & Mu, H. (2012). An oriented spanning tree based genetic algorithm for multi-criteria shortest path problems. *Applied Soft Computing, 12*, 506–512.
14. Hajjar, C., & Harndan, H. (2013). Interval data clustering using self organizing maps based on adaptive distance. *Neural Network, 11*(1), 124–132.
15. Bermejo, P., de la Ossa, L., Gamez, J. A., & Puerta, J. M. (2012). Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. *Knowledge-Based Systems, 1*(25), 35–44.
16. Kumari, A., Srinivas, K., & Gupta, M. (2013). Software module clustering using a hyper-heusristic based multi-objective genetic algorithm. In *Advance computing conferecne (IAAC), IEEE International Conference* (pp. 813–818).
17. Ferrira, A. J., Figueiredo, M. A. T., & Zhou, Z. H. (2012). Efficient feature selection filters for high-dimensional data. *Pattern Recognition Letters, 33*(13), 1794–1804.
18. Hsu, H. H., Hsieh, C. W., & Lu, M. D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications, 38*(7), 8144–8150.
19. Ginnakoglou, K. C., & Kampolis, I. C. (2010). *Multilevel optimization algorithms based on metamodel and fitness inheritance-assisted evolutionary algorithms. Computational intelligence in expensive optimization problems* (pp. 61–84). Berlin: Springer.
20. Deb, K. (2014). Multi-objective optimization. In E. K. Burke & G. Kendall (Eds.), *Search methodologies* (pp. 403–449). Berlin: Springer.
21. Anusha, M., & Sathiaseelan, J. G. R. (2015). An improved K-means genetic algorithm for multi-objective optimization. *International Journal of Applied Engineering Research, Special Issue, 10*(1), 228–231.
22. Antonio, L., & Coello Coello, C. (2013). Use of cooperative coevolution for solving large scale multi-objective optimization problems. In *IEEE Congress on Evolutionary Computation* (pp. 2758–2765). doi:10.1109/CEC.2013.6557903.
23. Garcia-Piquer, A., Fornells, A., Bacardit, J., Orriols-Puig, A., & Golobardes, E. (2014). Large-sclae experimental of cluster representation for multi-objective evolutionary clustering. *IEEE Transactions on Evolutionary Computation, 18*(1), 36–55.
24. Fieldsend, J. E., & Everson, R. M. (2015). The rolling tide evolutionary algorithm: A multiobjective optimizer for noisy optimization problems. *IEEE Transactions on Evolutionary Computation, 19*(1), 103–117.
25. Li, M., Yang, S., & Liu, X. (2014). Shift-based estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation, 18*(3), 348–365.
26. Lopez-James, A., & Coello, C. A. C. (2014). Including preferences into a many-objective engineering optimization problems. *Information Sciences, 277*, 1–20.
27. Gracia-Piquer, A., Fornells, A., Bacardit, J., Orriols, A., & Golobardes, E. (2014). Large-scale experimental evaluation of cluster representations for multiobjective evolutionary clustering. *IEEE Transactions on Evolutionary Computation, 18*(1), 36–53.
28. Wang, J., Zhong, C., Zhou, Y., & Zhou, Y. (2014). Multiobjective optimization algorithm with objective-wise learning for continuous multiobjective problems. *Journal of Ambient Intelligence and Humanized Computing, 6*, 571–585.
29. Wang, R., Fleming, P. J., & Purshouse, R. C. (2014). General framework for localised multi-objective evolutionary algorithms. *Information Sciences, 258*, 29–53.
30. Liu, H.-L., Gu, F., & Zhang, F. (2014). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation, 18*, 450–455.
31. Mukhopadhyay, A., & Bandyopadhyay, S. (2014). A survey of multiobjective evolutionary algorithms for data mining: Part-II. *IEEE Transactions on Evolutionary Computation, 18*(1), 20–35.
32. Coello, C. A. C. (2015). *Multi-objective evolutionary algorithms in real-world applications: Some recent results and current challenges*. Berlin: Springer.

33. Li, K., Fialho, A., Kwong, S., & Zhang, Q. (2014). Adaptive operator selection with bandits for a multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation, 18*(1), 114–130.
34. Carreno Jara, E. (2014). Multi-objective optimization by using evolutionary algorithms: The ρ-optimality criteria. *IEEE Transactions on Evolutionary Computation, 18*(2), 167–179.
35. Deb, K., & Jain, H. (2014). An evoutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part-I solving with box constraints. *IEEE Transactions on Evolutionary Computation, 18*(4), 577–601.
36. Deb, K., & Jain, H. (2014). An evoutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part-II: handing constraints and extending to an adaptive. *IEEE Transactions on Evolutionary Computation, 18*(4), 602–622.
37. Peng, P., Addam, O., Elzohbi, M., Ozyer, S. T., Elhajj, A., Gao, S., et al. (2014). Reporting and analyzing alternative clustering solutions by employing multi-objective genetic algorithm and conducting experiments. *Knowledge-Based Systems, 56*, 108–122.
38. Mukhopadhyay, A., Maulik, U., & Bandyopadhyay, S. (2013). An interactive approach to multiobjective clustering of gene expression patterns. *IEEE Transactions on Biomedical Engineering, 60*(1), 35–41.
39. Xia, H., Zhuang, J., & Dehong, Y. (2013). Novel soft subspace clustering with multi-objective evolutionary approach for high-dimensional data. *Pattern Recognition, 46*, 2562–2575.
40. Wang, Y., Xiang, J., & Cai, Z. (2012). A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator. *Applied Soft Computing, 12*, 3526–3538.
41. Bello-Orgaz, G., & Camacho, D. (2014). Evolutionary clustering algorithm for community detection using graph-based information. In *IEEE Congress on Evolutionary Computation* (pp. 930–937). doi:10.1109/CEC.2014.6900555.
42. Singh, H. K., Isaacs, A., & Ray, T. (2011). A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problem. *IEEE Transactions on Evolutionary Computation, 15*(4), 539–556.
43. Kirkland, O., Rayard-Smith, V. J., & de la Iglesia, B. (2011). *A novel multi-objective genetic algorithm for clustering* (pp. 317–326). Berlin: Springer.
44. Bandyopadhyay, S., Chakraborth, R., & Maulik, U. (2015). Priority based ε dominance: A new measure in multiobjective optmization. *Information Sciences, 305*, 97–109.
45. Liu, L., Mu, H., & Yang, J. (2015). Generic constraints handling techniques in constrained multi-criteria optimization and its application. *European Journal of Operational Research, 244*, 576–591.
46. Moritz, R. L. V., Reich, E., Schwarz, M., Bert, M., & Middendorf, M. (2014). Refined ranking relations for selections in multi-objective metaheuristics. *European Journal of Operational Research, 243*(2), 454–464.
47. Li, K., Kwong, S., & Deb, K. (2015). A dual-population paradigm for evolutionary multiobjective optimization. *Information Sciences, 309,* 50–72.
48. Inkaya, T., Kayalgil, S., & Ozdemirel, N. E. (2014). An adaptive neighbourhood construction algorithm based on density and connectivity. *Pattern Recognition Letters, 52,* 17–24.
49. Kimovski, D., Ortega, J., Ortiz, A., & Banos, R. (2015). Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection. *Expert Systems with Applications, 42*, 4239–4252.
50. Byers, C., Cheng, B. H. C., & Deb, K. (2015). *Unwanted feature interaction between the problem and search operators in evolutionary multi-objective optimization* (pp. 19–33). Berlin: Springer.
51. Ruiz, A. B., Luque, M., Miettinen, K., & Saborida, R. (2015). *An interactive evolutionary multiobjective optimization method: Interactive WASF-GA* (pp. 249–263). Berlin: Springer.
52. Saha, S., Spandana, R., Ekbal, A., & Bandyopadhyay, S. (2015). Simultaneous feature selection and symmetry based clustering using multiobjective framework. *Applied Soft Computing, 29*, 479–486.
53. Long, Q. (2014). A constraint handling technique for constrained multi-objective genetic algorithm. *Swarm and Evolutionary Computation, 15*, 66–79.
54. Long, Q., Wu, C., Huang, T., & Wang, X. (2015). A genetic algorithm for unconstrained multi-objective optimization. *Swarm and Evolutionary Computation, 22,* 1–14.
55. Chen, B., Zeng, W., Liu, Y., & Zhang, D. (2015). A new local search-based multi-objective optimization problem. *IEEE Transactions on Evolutionary Computation, 19*(1), 50–73.
56. Deb, K., Pratap, A., Agaral, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.
57. Dash, M., Liu, H., & Yao, L. (1997). Dimensionality reduction of unsupervised data. In *IEEE Conference in Artificial Intelligence* (pp. 532–539).

58. Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association, 336*(66), 846–850.
59. Jing, L. P., Ng, M. K., & Huang, Z. X. (2007). An entropy weighting k-means algorithm for subspace clustering of high dimensional sparse data. *IEEE Transactions on Knowledge and Data Engineering, 19*, 1026–1041.
60. http://www.ics.uci.edu/mlearn/MLRepository.html.

M. Anusha has received M.Sc., and M.Phil., degree in computer science in 2003 and 2014 respectively and M.Tech., degree in Bioinformatics from Sastra University. At present she is a full-time research scholar in the Department of Computer Science, Bishop Heber College, Trichy, Tamilnadu, India. She has published six papers in referred journals and conferences.



J. G. R. Sathiaseelan has received M.Sc., degree in computer science from Bharathidasan University, India, in 1989 and Ph.D., degree in computer science from Bharathidasan Univerity, India in 2013. He has more than 25 years of teaching experience in computer science. He is a Head, Department of computer science, Bishop Heber College (Autonomous), Tiruchirapalli, India. He authored a book on C# dot net and more than 30 papers have been published in referred journals and conferences proceedings in the area of web services. His current research interests include data mining, genetic algorithm, multi-objective optimization.