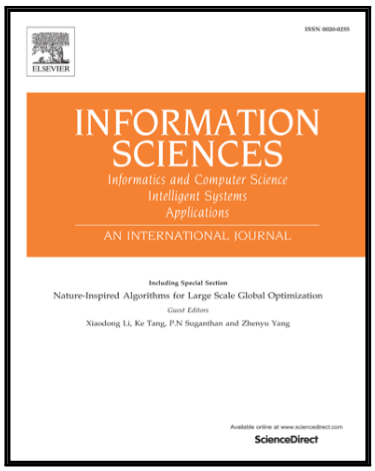


Multi-clustering via evolutionary multi-objective optimization

Rui Wang, Shiming Lai, Guohua Wu, Lining Xing, Ling Wang, Hisao Ishibuchi

PII: S0020-0255(18)30227-5
DOI: 10.1016/j.ins.2018.03.047
Reference: INS 13525

To appear in: *Information Sciences*

Received date: 17 October 2017
Revised date: 13 March 2018
Accepted date: 18 March 2018

Please cite this article as: Rui Wang, Shiming Lai, Guohua Wu, Lining Xing, Ling Wang, Hisao Ishibuchi, Multi-clustering via evolutionary multi-objective optimization, *Information Sciences* (2018), doi: 10.1016/j.ins.2018.03.047

Highlights

- The parallelism feature of evolutionary multi-objective optimization (EMO) can be used to search for multiple clustering results simultaneously.
- An *a posteriori* method, EMO-KC, is proposed to identify an appropriate cluster number.
- A transformation strategy is designed for the construction of bi-objective optimization problem.

Multi-clustering via evolutionary multi-objective optimization

Rui Wang[a,b], Shiming Lai[b], Guohua Wu[b], Lining Xing[b], Ling Wang[c], Hisao Ishibuchi[d,e]

[a] *School of Mathematics and Big Data, Foshan University, Foshan 528000, P.R.China*
[b] *College of Systems Engineering, National University of Defense Technology, Changsha, 410073, P.R. China*
[c] *Department of Automation, Tsinghua University, Beijing, 100084, P. R. China*
[d] *Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, 518055, P. R. China*
[e] *Department of Computer Science and Intelligent Systems, Osaka Prefecture University, Osaka 599-8531, Japan*

Abstract

The choice of the number of clusters (k) remains challenging for clustering methods. Instead of determining k, the implicit parallelism feature of evolutionary multi-objective optimization (EMO) provides an effective and efficient paradigm to find the optimal clustering in a *posteriori* manner. That is, first EMO algorithms are employed to search for a set of non-dominated solutions, representing different clustering results with different k. Then, a certain validity index is used to select the optimal clustering result. This study systematically investigates the use of EMO for multi-clustering (i.e., searching for multiple clustering simultaneously). An effective bi-objective model is built wherein the number of clusters and the sum of squared distances (SSD) between data points and their cluster centroids are considered as objectives. To ensure the two objectives are conflicting with each other, a novel transformation strategy is applied to the SSD. Then, the model is solved by an EMO algorithm. The derived paradigm, EMO-k-clustering, is examined on three datasets of different properties where NSGA-II serves as the EMO algorithm. Experimental results show that the proposed bi-objective model is effective. EMO-k-clustering is able to efficiently obtain all the clustering results for different k values in its single run.

Keywords: Multi-objective optimization, Evolutionary algorithms, Clustering.

1. Introduction

The era of information and big data enables data mining to become increasingly important in many areas such as internet search, finance, urban informatics, and business informatics. As an approach of knowledge discovery, the overall

Email address: ruiwangnudt@gmail.com (Rui Wang)

goal of data mining is to extract information from a data set and transform it into an understandable structure for further use. Its main task is to automatically or semi-automatically extract previously unknown, interesting patterns in large quantities of data sets. As data sets have grown in size and complexity, data mining methods are often aided by other approaches such as neural networks, evolutionary algorithms, decision trees, support vector machines and deep learning [10, 12].

Cluster analysis is one of the most important tasks in data mining, which has been widely applied in a variety of scientific areas such as pattern recognition, information retrieval, microbiology analysis, and so forth [16]. In general, a clustering method aims to partition n data points into k clusters. Unless providing a correct k value, the method will lead to inappropriate clustering results. Unfortunately, the choice of k is often application dependent [7]. Without *a prior* knowledge of how many clusters are really in the data, it is not easy to choose an appropriate value of k. Therefore, this study proposes to experiment with a range of values for k such that an user can flexibly choose a clustering result based on his/her preference or a certain validity index.

To obtain multiple clustering results, a natural way is to iteratively perform a standard clustering method, e.g., the k-means, with different k, each iteration for one clustering result. However, this is obviously deficient, in particular, when the size of data and/or the number of possible k values is large. Therefore, a novel clustering paradigm called EMO-k-clustering (EMO-KC for short) is proposed. The EMO-KC utilizes the implicit parallelism of EMO to synchronously obtain multiple clustering results in a single algorithm run. Specifically, the number of clusters k and the sum of squared distances between data points and their cluster centroids (SSD [8] which measures the compactness of the clustering) are considered as two objectives to be minimized. Since SSD and k are not always conflicting between two individuals, a novel transformation for SSD is proposed in this study which guarantees that the conflicting relationship holds for any two individuals having different k. The bi-objective model is then solved by an EMO algorithm (e.g., NSGA-II [4], MOEA/D [37, 33], PICEA [29, 30]). This results in a set of near-optimal trade-off solutions between SSD and k, representing different clustering results.

As an instance, in this study we use NSGA-II in EMO-KC. The proposed method is examined on three datasets with different properties, and is shown as effective. Given the ingenuity of the bi-objective model, EMO-KC[1] effectively finds all clustering results for all considered k values with only one execution of the algorithm. By further considering a clustering validity index, an appropriate clustering result is identified. In addition, experimental results show that EMO-KC achieves comparable clustering accuracy with a genetic algorithm-based clustering method, but significantly outperforms the latter in terms of computation time when multiple k values are required. Also, EMO-KC is shown to scale up well on high-dimensional datasets. Overall the main contributions

[1]For brevity EMO-KC refers to the use of NSGA-II hereafter.

of this study are i) the proposal of multi-clustering via EMO by which a preferred (an optimal) clustering result can be selected, and ii) a novel bi-objective formulation for EMO based multi-clustering.

The rest of this study is structured as follows. In Section 2, clustering methods and evolutionary multi-objective optimization are briefly explained. In Section 3, the proposed EMO-KC is elaborated. This is followed by an examination of the performance of EMO-KC in Section 4 and Section 5. Section 6 concludes this study and identifies some future studies.

2. Background

2.1. Clustering methods

A clustering method partitions a set of n data points, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ into k homogeneous clusters such that data points within a cluster are close to each other and far from those in different clusters. For example, the k-means starts with k initial cluster centroids, $\{\mathbf{m}_1, \mathbf{m}_1, \ldots, \mathbf{m}_k\}$. An input $\mathbf{x}_i$ is assigned into the jth cluster if the indicator function $I(j|\mathbf{x}_i) = 1$ holds with

$$I(j|\mathbf{x}_i) \begin{cases} 1, & \text{if } j = \arg\min_{1 \le r \le k} ||\mathbf{x}_i - \mathbf{m}_r||^2 \\ 0, & otherwise \end{cases} \tag{1}$$

That is, each data point is assigned to its closest centroid. All data points that are assigned to a specific cluster centroid constitute a cluster. The candidate cluster centroids are then updated, e.g., taking the mean of all data points assigned to the jth centroid as the new centroid. The above process repeats till a stopping criterion is met, e.g., all cluster centroids converge.

2.2. Determination of an optimal k

Since determining k *a priori* is difficult, a natural question arises: what the optimal k should be in order to obtain well defined clusters. In literature, there have been numerous studies proposed to deal with this issue [16, 7]. One class of the methods is known as automatic data clustering. That is, first, one or multiple clustering criteria are designed for evaluation of the clustering results. Second, an algorithm is performed to optimize the criteria wherein k is considered as a decision variable. For example, in [8] evolutionary algorithms are applied to optimize the clustering criteria for simultaneously determining the cluster number as well as clustering the data objects. Another class of methods can be referred to as *posteriori* methods. That is, multiple clustering results for different k are first obtained, then a certain cluster validity index is applied to evaluate the clustering results. The one that gives the best index value is selected.

Posteriori methods are our concern in this study. To the best of our knowledge, most of *posteriori* methods focus on the design of validity index or the use of various validity indices to perform clustering. With respect to obtaining multiple clustering results under different k, existing studies often employ

the naive idea, that is, performing a clustering method iteratively with a range of k values. We are aware of only few studies [15, 18], exploring the idea of multi-clustering via evolutionary multi-objective optimization. In [15, 18], the total-within-cluster-variation (TWCV) and k are directly used as objective functions (which will be shown as ineffective later in this study). The bi-objective model is then solved by an EMO algorithm, NPGA [9]. In addition, the cluster label-based encoding is employed [15, 18], i.e., the length of a chromosome is equal to the number of points in the dataset, and each position denotes the cluster label of the respective point. The main advantage of this type of encoding is that the decoding step is straightforward, and it looks suitable for any data types. However, there are also disadvantages, e.g., the chromosome length is the same as the number of points. This may create difficulty for algorithm convergence.

2.3. Evolutionary Multi-objective optimization

Evolutionary multi-objective optimization is to solve multi-objective problems (MOPs) by evolutionary algorithms. MOPs refer to problems that have multiple objective functions to be simultaneously optimized, see Eq. (2).

$$\begin{aligned} \text{Min} \quad & F(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})\} \\ \text{such that} \quad & \mathbf{x} = (x_1, x_2, \ldots, x_i, \ldots, x_n) \in \Omega \end{aligned} \tag{2}$$

where Ω denotes the search space, m is the number of objectives, $\mathbf{x}$ is the decision vector consisting of n decision variables x_i. A solution $\mathbf{x}$ is said to Pareto dominate another solution $\mathbf{y}$, if and only if, $\forall i = 1, 2, ..., m, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ $\wedge \exists j = 1, 2, ..., m, f_j(\mathbf{x}) < f_j(\mathbf{y})$. Furthermore, a Pareto optimal solution is the one that is not Pareto dominated by any other solutions. The image of all Pareto optimal solutions in the objective space is termed the Pareto optimal front.

Since objective functions in MOPs are often in conflict with one another, the optimal solution of MOPs is not a single one but rather a set of non-dominated solutions [3]. These solutions present different trade-offs between objectives. The decision-maker can choose a solution based on his/her preference. Evolutionary multi-objective algorithms, e.g., [4, 37, 36] are well-suited for solving MOPs [23] since their population based nature enables to generate multiple trade-off solutions in a single algorithm run. These trade-off solutions are expected to be as close as possible to (convergence), and as evenly (uniformity) and widely (diversity) distributed along the entire Pareto optimal front as possible [11, 32]. In the last two decades, a number of EMO algorithms have been proposed. To name some representatives, Pareto dominance-based algorithms, e.g., [4], decomposition based algorithms, e.g.,[4], indicator based algorithms, e.g., [20, 21], preference based algorithms [29, 28]. In addition, EMO algorithms are often applied to aid decision-making [31, 22].

3. Evolutionary Multi-clustering Optimization: EMO-k-clustering

This section elaborates how multiple clustering results are obtained via evolutionary multi-objective optimization. The procedure contains mainly two steps [5, 34]: i) constructing two conflicting objective functions, and ii) solving the bi-objective optimization problem with an effective EMO algorithm.

3.1. Bi-objective model

The intra-cluster distance is measure of the sum of squared distances (SSD) between data points and their cluster centroids, see Eq. (3). Minimizing the sum of squared distances shows homogeneity and tightness of the cluster. The SSD in general decreases as k increases. To the extreme case, i.e., k is the same as the number of data points, the SSD reaches zero. Thus, the two conflicting objective functions are constructed based on the two measures. Specifically, they are defined as follows.

$$\begin{aligned} \text{Min} \quad & F(\mathbf{x}) = \{f_1(\mathbf{x}) = (1 - \exp^{-1 \cdot \text{SSD}}) - k, f_2(\mathbf{x}) = k\} \\ \text{where} \quad & \text{SSD} = \sum_{r=1}^{k} \sum_{\mathbf{x}_i \in C_r} ||\mathbf{x}_i - \mathbf{m}_r||^2 \\ & \mathbf{m}_r = (m_r^1, m_r^2, \ldots, m_r^d) \end{aligned} \tag{3}$$

where $\mathbf{m}_r = (m_r^1, m_r^2, \ldots, m_r^d)$ denotes the rth cluster centroid, d is the dimensionality of a data object, i.e., the number of features describing a data object. C_r denotes the collection of data points in the rth cluster. Other validity indices are also applicable, e.g., the total-within-cluster-variation (TWCV [16]). However, since Eq. (3) requires the index value SSD ≥ 0, the employed validity index should be slightly modified by subtracting the minima of the index values.

The reason that f_1 is not directly set as SSD, i.e., $f_1(\mathbf{x}) \neq \text{SSD}$ is as follows. The monotonic decreasing property of SSD (as k increases) holds only if the true cluster centroids are found [8]. Before approaching to the optimal case (true cluster centroids), the conflicting relationship between SSD and k is not guaranteed, see Figure 1 for an illustration.

Therefore, if the bi-objective model is built with $\{f_1(\mathbf{x}) = \text{SSD}, f_2(\mathbf{x}) = k\}$, it would be very likely that solutions for some k values are dominated during the search. In the worst case, no solution will be found for those k values at the end. It is easy to know that this observation also applies to other cluster validity indices such as the TWCV metric used in [15, 18]. A similar illustration is provided in Appendix A.

According to the above analysis, in Eq. (3) a transformation is applied to the SSD. By the transformation $1 - \exp^{-1 \cdot \text{SSD}} - k$, f_1 and f_2 are guaranteed to be conflicting for any two solutions having different k values.

Proof. Assuming that $\mathbf{s}_1$ and $\mathbf{s}_2$ are two randomly selected solutions, and their assigned k values are k_1 and k_2, respectively. Without loss of generality, we

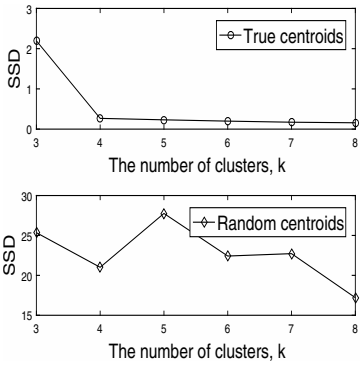


Figure 1: SSD values with true and rand cluster centroids over different k.

assume $k_1 > k_2$. Thus,

$$f_2(\mathbf{s}_1) - f_2(\mathbf{s}_2) = k_1 - k_2 \geq 1$$

Meanwhile, $f_1(\mathbf{s}_1) - f_1(\mathbf{s}_2) =$

$$\begin{aligned} &(1 - \exp^{-1\cdot \mathrm{SSD}(\mathbf{s}_1, k_1)} - k_1) - (1 - \exp^{-1\cdot \mathrm{SSD}(\mathbf{s}_2, k_2)} - k_2) \\ =&(k_2 - k_1) - (\exp^{-1\cdot \mathrm{SSD}(\mathbf{s}_2, k_2)} - \exp^{-1\cdot \mathrm{SSD}(\mathbf{s}_1, k_1)}) \end{aligned} \tag{4}$$

Since $\exp^{-1\cdot \mathrm{SSD}}$ is always within $(0, 1)$, so $f_1(\mathbf{s}_1) - f_1(\mathbf{s}_2) < 0$. This ends the proof. □

Next we prove that every Pareto optimal solution of Eq. (3) corresponds to an optimal solution of $\mathrm{Min} f(\mathbf{x}, k) = \mathrm{SSD}$, i.e., the optimal clustering result for a certain k.

Proof. Assuming that (obj_1, obj_2) is a solution on the Pareto optimal front (and its associated decision vector is $\mathbf{x}'$. Therefore, when $f_2 = obj_2$ (a certain cluster number), there is no solution that can produce a smaller f_1 than obj_1, thus obj_1 is minimum, i.e., $f_1(\mathbf{x}')$ is minimum. Since the second part of $f_1(\mathbf{x}')$ is a constant, so $1 - \exp^{-1\cdot \mathrm{SSD}}$ is minimum for the case $k = obj_2$ and the optimal solution of $\mathrm{Min} f(\mathbf{x}, k) = \mathrm{SSD}$ is $\mathbf{x}'$.

This ends the proof. □

3.2. Optimizer

With respect to the optimizer, the EMO algorithm– NSGA-II [4] is chosen due to its simpleness and robust performance on two-objective problems, though other EMO algorithms can also be used. Specifically, the NSGA-II is slightly tailored to solve the constructed two-objective problem. Its pseudo-code is shown in Algorithm 1.

The derived algorithm employs an $(\mu + \mu)$ elitist framework as shown in Figure 2. It starts with a set of N randomly generated parent solutions. At

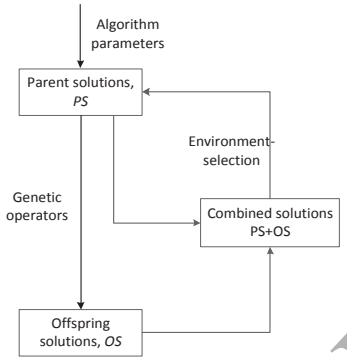


Figure 2: Illustration of the $(\mu + \mu)$ elitist framework in NSGA-II.

Algorithm 1: The tailored NSGA-II for multi-clustering.

Input: Maximum generation $maxGen$, population size N, a range of k values

Output: PS

Initialize a set of N random solutions, PS;
Assign each solution with a random different value of k;
while $gen \leq maxGen$ **do**
    Generate N offspring solutions OS by crossover and mutation operators;
    Combine PS and OS together to form $jointS$;
    Evaluate $jointS$ by the fast non-dominated sorting approach and the crowding distance [4];
    Select the best N solutions from $jointS$ to form the new parent PS;
    $gen \leftarrow gen + 1$;
end

each iteration, the same number of offspring are produced through selection, crossover and mutation operators (e.g., simulated binary crossover (SBX) and polynomial mutation (PM) [4]). Then parent solutions and their offspring are combined to form a joint population. Solutions in the joint population are then ranked by the fast non-dominated sorting approach (based on the Pareto dominance relation). Amongst equally ranked solutions, the secondary criterion, crowding distance, is employed to select solutions in less crowded regions so as to enhance the diversity of solutions. Subsequently, N solutions are selected from the joint population as the new parents. Next we elaborate the solution encoding, crossover and mutation operators used in this algorithm. All the other components adopted in the algorithm are the same as the original NSGA-II.

3.2.1. Solution encoding

With respect to the solution encoding strategy, the centroid-based encoding is adopted [16]. The individual (chromosome) is composed of real numbers that represent the coordinates of the cluster centroids. Moreover, in order to handle different number of clusters, a unified chromosome is applied. That is, all chromosomes are initialized with the length of $d \cdot k_{max}$ where d is the dimensionality of data points and k_{max} is the maximum k value. The default range of k is $[1, k_{max}]$. After the initialization, each chromosome is assigned with a random k. Therefore, during the search, only the first $d \cdot k$ genes are taken as decision variables for a chromosome. The range of each gene (the cluster centroid) is bounded by the lower bound and upper bound of the datasets. For example, assuming that we have a chromosome $\mathbf{s}_1$=(0.5,0.3,0.4,0.1,0.7,0.8,0.2,0.6) and $k_{max} = 4$, $d = 2$. Therefore, when $k = 2$ is assigned to $\mathbf{s}_1$, only (0.5,0.3,0.4,0.1) will be taken as decision variables.

3.2.2. Crossover and mutation

For two randomly selected individuals, $\mathbf{s}_1$ and $\mathbf{s}_2$, the SBX and PM operators are applied to produce new offspring. The SBX operator is a two-parent variation operator that produces two new solutions. It has two controllable parameters: i) the probability of applying recombination p_c to a pair of parent solutions, and ii) the magnitude of the expected variation from the parent values, (η_c). The PM operator also has two controllable parameters: i) the probability of applying mutation (p_m), and ii) a mutation distribution parameter (η_m). $\mathbf{s}'_1$ and $\mathbf{s}'_2$ inherent k values of $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. Note that the crossover is carried out across all solutions, rather than being restricted only within solutions having the same k. This is because performing crossover between solutions with different k values may facilitate knowledge transfer which can increase the exploration ability further [6]. The effect is discussed in Section 5.2.

Lastly, it is worth mentioning that multi-populations based evolutionary algorithms (MPEAs [25, 24]) can also be applied to obtain multiple clustering results simultaneously. MPEAs evolve multiple populations during the search, each population for a single k setting. However, MFEAs may not be suitable when k is large [13]. Nevertheless, EMO-KC can be benefited from the MPEA framework since MPEAs are inherently suitable for parallel computing. That is,

the efficiency of EMO-KC could be improved even further based on the parallel mode of MPEAs.

4. Experiments

4.1. Test datasets and algorithm parameters

Experimental datasets are three simulated artificial datasets which are generated using normal distribution with different values of parameters– mean (μ) and standard deviation (δ), as shown in Table 1. The first dataset, denoted as DS_100_4 is relatively simple which contains 100 objects, and forms four clear clusters with equal size. The second dataset, denoted as DS_500_6, contains 500 objects, and forms six clusters with different sizes. The third dataset, denoted as DS_900_7, contains 900 objects, and forms seven clusters. Some outliers are added to increase the difficulty of clustering task. The lower and upper bounds of data objects are [0,0] and [1,1], respectively. Here, only two-dimensional datasets are shown since their clustering results can be easily visualized. Multi-dimensional datasets will be studied in Section 5.

Table 1: Features of artificial datasets (where n_O = number of objects, n_C = number of clusters, μ= mean, δ = standard deviation).

DS_100_4 $n_O = 100$ $n_C = 4$	$\mu_1 = [0.2, 0.2]$, $\mu_2 = [0.2, 0.6]$ $\mu_3 = [0.6, 0.2]$, $\mu_4 = [0.6, 0.6]$	δ = [0.001, 0; 0, 0.002]
DS_500_6 $n_O = 500$ $n_C = 6$	$\mu_1 = [0.2, 0.2]$, $\mu_2 = [0.2, 0.6]$ $\mu_3 = [0.6, 0.2]$, $\mu_4 = [0.6, 0.6]$ $\mu_5 = [0.4, 0.4]$, $\mu_6 = [0.4, 0.8]$	δ = [0.001, 0; 0, 0.002]
DS_900_7 $n_O = 900$ $n_C = 6$	$\mu_1 = [0.1, 0.2]$, $\mu_2 = [0.4, 0.2]$ $\mu_3 = [0.8, 0.2]$, $\mu_4 = [0.2, 0.5]$ $\mu_5 = [0.7, 0.6]$, $\mu_6 = [0.1, 0.8]$ $\mu_7 = [0.6, 0.9]$	δ = [0.003, 0; 0, 0.005]

Parameters of EMO-KC are shown in Table 2. These settings are kept constant for all algorithm runs.

Table 2: General parameter settings

maxGen	N	SBX	PM
250	90	$p_c = 1, \eta_c = 15$	$p_m = 1/n, \eta_m = 20$

4.2. Experimental results

4.2.1. The performance of EMO-KC

Though a wider range of k values can be considered, here k is set within the interval [3,20] for illustration. The non-dominated solutions[2] for each $k \in [3, 20]$ (in the objective space) obtained by the EMO-KC are shown in Figure 3.

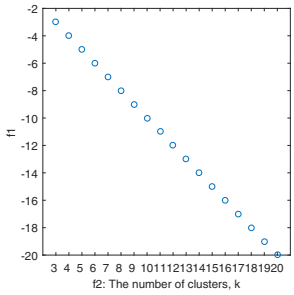


Figure 3: Pareto fronts of DS_900_7 obtained by EMO-KC. Results for the other two datasets are similar, and thus, are not shown here.

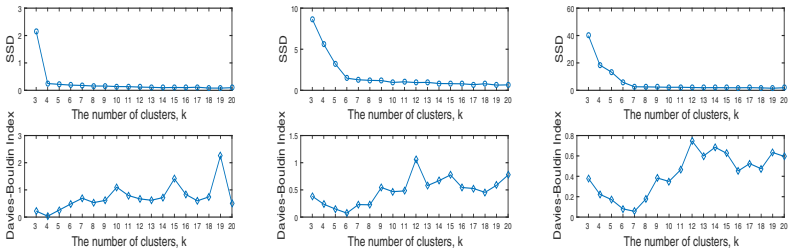
From Figure 3, it can be observed clearly that 18 non-dominated solutions, representing clustering results for all considered k values, are obtained for the DS_900_7 datasets. Next we show how the best clustering result (the optimal k) is selected by the "elbow" method [7]. In the method, SSD against k is plotted. The "elbow point" where the rate of decrease sharply shifts is considered as the optimal k. Since there could be more than one elbow, or no elbow at all for some datasets, the Davies-Bouldin (DB) index [16] which is defined as the ratio of the sum of the within cluster dispersions to the between cluster separation is further considered to determine the best clustering result (corresponding to the optimal k). The smaller the DB index the better the clustering result.

The "elbow plot" as well as the DB index for the three datasets are shown in Figure 4. From the figure we can clearly observe that there are elbow points for the three datasets, which are $k = 4, 6, 7$, respectively. Moreover, according to the DB index, one can also find that $k = 4, 6, 7$ provide the best clustering results for the three datasets, respectively.

Having determined the optimal k, we then evaluate how well the datasets are clustered. The number of wrongly clustered data points is counted, and is used as a metric. It is found that for all the three datasets none of data points is wrongly clustered, which indicates that EMO-KC is effective.

Essentially, as the EMO-KC adopts the centroid-based encoding, it may encounter similar issues as the k-means. For example, EMO-KC can handle sphere-shaped clusters well, but may not capture clusters with non-convex and/or in-

[2]Statistically, the solution set with the median hypervolume [38] value across 31 algorithm runs is shown. Hypervolume (HV) is a widely used performance indicator to evaluate the performance of EMO algorithms. A larger HV implies both good convergence and diversity [38].



(a) DS_100_4, $k^* = 4$ (b) DS_500_6, $k^* = 6$ (c) DS_900_7, $k^* = 7$

Figure 4: Elbow plot and the DB index over k for the three datasets.

consistent shapes. Nevertheless, this issue can be circumvented by employing other encodings in EMO-KC. Overall, the main superiority of EMO-KC compared to k-means is the elimination of determining k *a priori*. Besides, EMO-KC provides multiple clustering results against different k, which might be useful for a decision-maker who can select a suitable clustering result based on his/her preference. Lastly, as EMO-KC uses evolutionary algorithms as search engine, it is not easy to get trapped into local optima as the k-means.

4.2.2. Comparison of EMO-KC and GA-KC

This section demonstrates the superiority of EMO-KC over a genetic algorithm-based k-clustering (denoted as GA-KC). To enable a fair comparison, the two methods take the same population size, the number of generations, crossover and mutation operators, see Table 2. The only difference is that GA-KC is a single-objective optimizer wherein only the first objective in Eq. (3) is optimized. Moreover, when initializing individuals in GA-KC, the number of k is assumed to known. This means that the length of an individual in GA-KC is smaller than the EMO-KC.

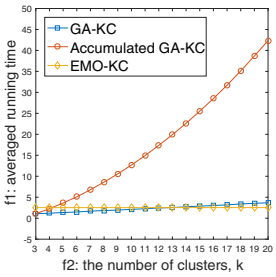


Figure 5: Averaged run time of EMO-KC and GA-KC on DS_900_7 for 250 generations. Both methods are implemented on a workstation with an Intel Core i7-4600U CPU at 2.10 GHz and 8 GB RAM running the Windows 7 operating system.

Figure 5 shows the run time of EMO-KC and GA-KC on DS_900_7 for different values of k from 3 to 20. Since EMO-KC obtains all results in a

single run, its run time is theoretically constant which is 2.5271(s) despite the stochastic feature of operation systems. The GA-KC has to run 18 times so as to obtain all results. Thus, its overall run time is accumulated which is 42.3504(s). To be statistical, the running time are averaged across 31 runs for both methods.

It is observed that from Figure 5 that EMO-KC consumes less time than GA-KC if more than two k values are considered. Therefore, if the number of clusters can be determined *a priori*, GA-KC is recommended. However, if one has to make more guesses about k, then EMO-KC is more efficient.

Moreover, the accuracy of the clustering results (in terms of the number of wrongly clustered data objects) by the two methods is also examined. Both the methods can correctly identify the optimal k. Also, all data points are appropriately clustered.

4.2.3. Comparison of EMO-KC with/without the transformation

To investigate the effect of the transformation strategy– ensuring the conflicting relationship between objective functions, a comparative study, i.e., the bi-objective model with and without the transformation, is conducted. The EMO-KC with the same parameter settings is applied to the three datasets. The obtained Pareto front for the dataset DS_900_7 is shown in Figure 6 for an instance.

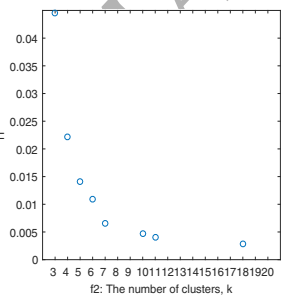


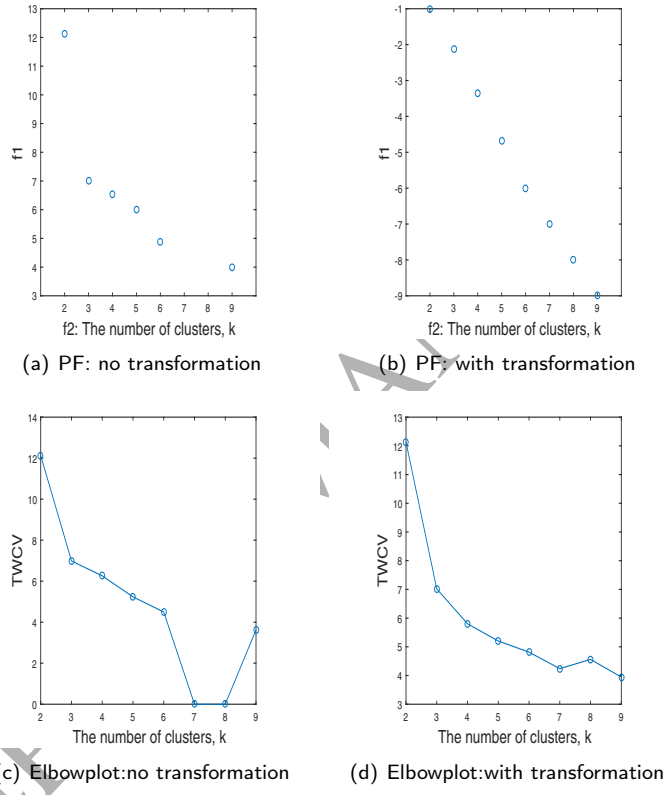
Figure 6: Pareto fronts of DS_900_7 obtained by EMO-KC without the transformation

From Figure 6 we can clearly observe that without the transformation, no solution is found for some k settings, e.g., $k = 8, 12, 19, 20$. This clearly demonstrates that our proposed transformation is effective. Although only the results for DS_900_7 are shown, similar results are observed for other datasets.

4.2.4. Comparison of EMO-KC with MOKGA

In this section we further demonstrate the advantages of EMO-KC by comparing it against an earlier work, MOKGA [15, 18] on the well-known dataset, Iris. The Iris dataset contains 150 instances that are from three classes (setosa, versicolour, virginica). There are 50 instances in each class. Each instance is described with four features, sepal length, sepal width, petal length, petal width.

To make a fair comparison study, in both EMO-KC and MOKGA, the centroid-based clustering, the same SBX and PM variation operator are adopted. Also, both methods take the same population size and maximum number of generation. Specifically, in EMO-KC the objectives are minimization of the transformed TWCV and the number of clusters while in MOKGA the objectives are minimization of TWCV metric and the number of clusters. Definition of TWCV is shown in Appendix A. Comparison results are show in Figure 7.



(a) PF: no transformation (b) PF: with transformation

(c) Elbowplot:no transformation (d) Elbowplot:with transformation

Figure 7: Comparison results between EMO-KC and MOKGA.

From the results we can observe that both the methods can find the optimal number of clusters, i.e., k=3. However, the difference is that EMO-KC is able to return all the specified clustering results while MOKGA cannot. The MOKGA returns no result for k=7 and 8. It is argued that although for the IRIS data MOKGA has found the optimal clustering however, it may face difficulty for other datasets since it cannot find all the required clustering results, i.e., results for all the specified k values. Moreover, the clustering accuracy is also calculated, 7.5 instances (averaged 30 algorithm runs) are wrongly classified in EMO-KC while 8.2 instances (averaged 30 algorithm runs) are wrongly classi-

fied in MOKGA. Therefore, we can draw a conclusion that EMO-KC is more robust than MOKGA.

5. Discussion

5.1. Scalability of EMO-KC on high-dimensional datasets

This section examines the scalability of EMO-KC on high-dimensional datasets. Five datasets with two, three, five, seven and nine dimensions are considered respectively. Again, k is assumed to be in the interval $[3, 20]$. Elbow plots for all datasets are shown in Figure 8.

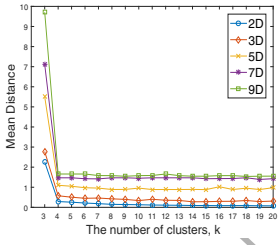


Figure 8: Elbow plots of dataset DS_100_4 with different dimensions.

From the results we can tentatively conclude that EMO-KC scales up well on high-dimensional datasets. Since the number of decision variables in EMO-KC increases with the dimensionality of the data point, clustering problems become large-scale optimization [35] when the number of attributes is hundreds or thousands. One can employ cooperative co-evolution strategies to further improve the performance of EMO-k-means on high dimension datasets [17, 19].

5.2. Cooperation between k and its neighbours

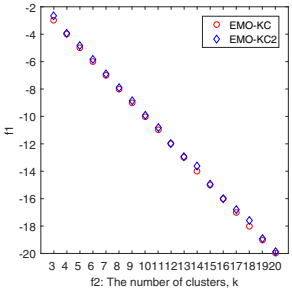


Figure 9: Illustration of Pareto fronts obtained by EMO-KC and EMO-KC2 for DS_100_4.

Empirical results have demonstrated both the effectiveness and efficiency of EMO-KC. We think that the advantage of EMO-KC is because of the implicit

cooperation amongst neighbouring solutions. That is, the crossover operation is performed within the entire population rather than only within solutions associated with the same k. To verify this hypothesis, EMO-KC is compared with EMO-KC2 in which crossover is allowed only for solutions associated with the same k. Comparison results (the obtained Pareto front) for DS_100_4 are shown in Figure 9 from which we can clearly see that almost all solutions obtained by EMO-KC2 are Pareto dominated by those obtained by EMO-KC. Thus, EMO-KC clearly outperforms EMO-KC2. Note that similar results are observed for other datasets.

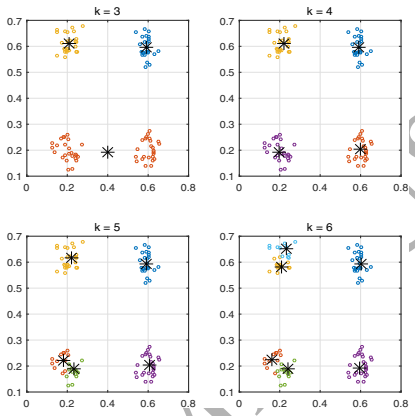


Figure 10: Cluster centroids over different k. Although only results for DS_100_4 with $k \in [3, 6]$ are shown, similar observations are obtained for the other two datasets.

Given a further thought, the obtained non-dominated solutions for different k are very likely to share common features. Some cluster centroids in a non-dominated solution with k clusters may be similar to those in a non-dominated solution with $k+1$ or $k-1$ clusters. Taking the dataset DS_100_4 as a test instance, Figure 10 shows the obtained cluster centroids ($\star$) for $k = 3, 4, 5, 6$. As is expected, some centroids are almost the same for different k. This also indicates that performing crossover within the entire population (amongst different k values) is helpful.

6. Conclusion

Determining an appropriate k *a priori* for data clustering is a long-standing question. This study, without pre-determining k, proposed to harness the implicit parallelism of EMO for multi-clustering. That is, first searching for clustering results for a range of different k values, then selecting the optimal clustering based on a certain clustering validity index. In contrast to existing studies that different clustering results are obtained iteratively, the proposed EMO based clustering method obtains all clustering results in its single run. As an implementation of the idea, the NSGA-II based EMO-k-clustering is tested on

three datasets of different properties, and is demonstrated as both effective and efficient. Moreover, in EMO-KC a simple yet effective objective transformation strategy is developed, which is demonstrated as helpful in enhancing the algorithm performance.

With respect to future studies, the first is to examine EMO-KC on more complicated, and real datasets. Second, instead of taking SSD as the basis of objective function, other validity indices can be applied. Moreover, it is known that validity indices could be sensitive to structures of datasets, a hybrid validity indices could be considered. Third, the transformation strategy is shown as important for multi-clustering, though it seems to have been overlooked in literature. Thus, more effective transformation strategies would be investigated. Fourth, effective EMO algorithms [1, 2, 14, 27, 20, 21] can be developed to aid the large scale optimization arise in data clustering. Lastly, it is worth mentioning that multi-clustering effectively is an instance of multi-cardinality constrained optimization [26] (or evolutionary multitasking [6]). The proposed method therefore provides new insights for such problems.

Overall, the parallelism of evolutionary multi-objective optimization has shown promising for searching for multiple clustering results simultaneously. However, this concept is still in its infancy, more rigorous studies are needed in the future. Source code of EMO-k-clustering is available at `http://ruiwangnudt.gotoip3.com/optimization.html`.

Appendix A. The TWCV value over different k

It has been shown in Figure 1 that SSD does not monotonically decrease as k increases. Since many studies also employ the total-within-cluster-variation (TWCV) as cluster validity index [16], the necessity of applying the transformation strategy to TWCV is demonstrated in this section. Similarly, the TWCV value with true and rand cluster centroids over different k is presented in Figure A.11. The TWCV index is defined as follows.

$$\text{TWCV} = \sum_{i=1}^{n}\sum_{j=1}^{d} x_{ij}^2 - \sum_{k=1}^{K}\frac{1}{n_k}\sum_{j=1}^{d}\left(\sum_{x_i\in C_k} x_{ij}\right)^2 \tag{A.1}$$

where x_{ij} denotes the $j$the feature value of the ith data point, and n_k denotes the number of points in cluster C_k.

It can be clearly observed that the monotonic decreasing property does not hold for TWCV when the true centroids are not found. Thus, it is recommended to apply the transformation strategy when TWCV is taken as an objective in [15, 18].

Acknowledgment

This work was supported by the National key research and development plan (2016YFB0901900), the National Natural Science Foundation of China

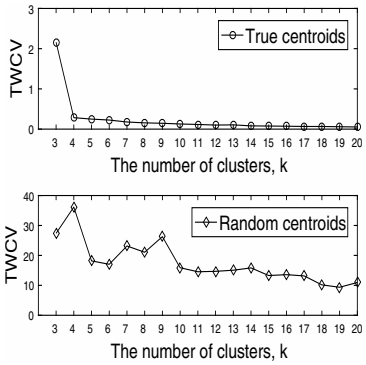


Figure A.11: TWCV values with true and rand cluster centroids over different k for $Data\_100\_4$ dataset.

(Nos. 61773390, 71371067 and 71571187) and the Distinguished Natural Science Foundation of Hunan Province (2017JJ1001). This work was also supported by JSPS KAKENHI Grant Numbers 16H02877 and 26540128.

References

[1] X. Cai, X.Z. Gao, Y. Xue, Improved bat algorithm with optimal forage strategy and random disturbance strategy, International Journal of Bio-inspired Computation 8 (2016) 205–214.

[2] X. Cai, H. Wang, Z. Cui, J. Cai, Y. Xue, L. Wang, Bat algorithm with triangle-flipping strategy for numerical optimization, International Journal of Machine Learning & Cybernetics (2017) 1–17.

[3] C. Coello, G. Lamont, D. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007.

[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[5] W. Gong, Y. Wang, Z. Cai, S. Yang, A weighted biobjective transformation technique for locating multiple optimal solutions of nonlinear equation systems, IEEE Transactions on Evolutionary Computation PP (2017) 1–1.

[6] A. Gupta, Y.S. Ong, L. Feng, Multifactorial evolution: Toward evolutionary multitasking, IEEE Transactions on Evolutionary Computation 20 (2016) 343–357.

[7] E. Hancer, D. Karaboga, A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number, Swarm and Evolutionary Computation 32 (2017) 49–67.

[8] J. Handl, J. Knowles, An Evolutionary Approach to Multiobjective Clustering, IEEE Transactions on Evolutionary Computation 11 (2007) 56–76.

[9] J. Horn, N. Nafpliotis, D. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, in: IEEE Congress on Evolutionary Computation (CEC), IEEE, pp. 82–87.

[10] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, A survey of evolutionary algorithms for clustering, IEEE Transactions on Systems Man & Cybernetics Part C 39 (2009) 133–155.

[11] H. Ishibuchi, T. Yoshida, T. Murata, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, IEEE Transactions on Evolutionary Computation 7 (2003) 204–223.

[12] A.K. Jain, Data Clustering: 50 Years Beyond K-means, Pattern Recognition Letters 31 (2010) 651–666.

[13] C. Li, T.T. Nguyen, M. Yang, M. Mavrovouniotis, S. Yang, An adaptive multipopulation framework for locating and tracking multiple optima, IEEE Transactions on Evolutionary Computation 20 (2016) 590–605.

[14] Y. Li, F. Liu, A novel immune clonal algorithm., IEEE Transactions on Evolutionary Computation 16 (2012) 35–50.

[15] Y. Liu, T. Özyer, R. Alhajj, K. Barker, Integrating Multi-Objective Genetic Algorithm and Validity Analysis for Locating and Ranking Alternative Clustering, Informatica 29 (2005) 33–40.

[16] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, A Survey of Multiobjective Evolutionary Clustering, Acm Computing Surveys 47 (2015) 1–46.

[17] M.N. Omidvar, X. Li, Y. Mei, X. Yao, Cooperative Co-Evolution With Differential Grouping for Large Scale Optimization, IEEE Transactions on Evolutionary Computation 18 (2014) 378–224.

[18] T. Özyer, Y. Liu, R. Alhajj, K. Barker, Multi-objective Genetic Algorithm Based Clustering Approach and Its Application to Gene Expression Data, Multi-objective Genetic Algorithm Based Clustering Approach and Its Application to Gene Expression Data, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 451–461.

[19] X. Peng, Y. Wu, Large-scale cooperative co-evolution using niching-based multi-modal optimization and adaptive fast clustering, Swarm and Evolutionary Computation 35 (2017) 65 – 77.

[20] S. Rostami, F. Neri, Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm, Integrated Computer Aided Engineering 23 (2016) 1–17.

[21] S. Rostami, F. Neri, A fast hypervolume driven selection mechanism for many-objective optimisation problems, Swarm and Evolutionary Computation 34 (2017) 50–67.

[22] S. Rostami, F. Neri, M. Epitropakis, Progressive preference articulation for decision making in multi-objective optimisation problems, Integrated Computer Aided Engineering (2017) 1–21.

[23] R. Shang, B. Du, H. Ma, L. Jiao, Y. Xue, R. Stolkin, Immune clonal algorithm based on directed evolution for multi-objective capacitated arc routing problem, Applied Soft Computing 49 (2016) 748–758.

[24] R. Shang, Y. Wang, J. Wang, L. Jiao, S. Wang, L. Qi, A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem, Information Sciences 277 (2014) 609–642.

[25] W. Sheng, S. Chen, M. Sheng, G. Xiao, J. Mao, Y. Zheng, Adaptive multisubpopulation competition and multiniche crowding-based memetic algorithm for automatic data clustering, IEEE Transactions on Evolutionary Computation 20 (2016) 838–858.

[26] R. Stephan, Cardinality constrained combinatorial optimization: Complexity and polyhedra, Discrete Optimization 7 (2010) 99–113.

[27] R. Wang, H. Ishibuchi, Z. Zhou, T. Liao, T. Zhang, Localized weighted sum method for many-objective optimization, IEEE Transactions on Evolutionary Computation 22 (2016) 3–18.

[28] R. Wang, R.C. Purshouse, P.J. Fleming, Preference-inspired coevolutionary algorithm using adaptively generated goal vectors, in: IEEE Congress on Evolutionary Computation (CEC), IEEE Press, Piscataway, NY, USA, 2013, pp. 916–923.

[29] R. Wang, R.C. Purshouse, P.J. Fleming, Preference-inspired Coevolutionary Algorithms for Many-objective Optimisation, IEEE Transactions on Evolutionary Computation 17 (2013) 474–494.

[30] R. Wang, R.C. Purshouse, P.J. Fleming, Preference-inspired coevolutionary algorithms using weight vectors, European Journal of Operational Research 243 (2015) 423–441.

[31] R. Wang, R.C. Purshouse, I. Giagkiozis, P.J. Fleming, The iPICEA-g: a new hybrid evolutionary multi-criteria decision making approach using the brushing technique, European Journal of Operational Research 243 (2015) 442–453.

[32] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, T. Zhang, On the effect of reference point in MOEA/D for multi-objective optimization, Applied Soft Computing 58 (2017) 25–34.

[33] R. Wang, Q. Zhang, T. Zhang, Decomposition based algorithms using Pareto adaptive scalarizing methods, IEEE Transactions on Evolutionary Computation 20 (2016) 821–837.

[34] Y. Wang, H.X. Li, G.G. Yen, W. Song, MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems, IEEE transactions on cybernetics 45 (2015) 830–843.

[35] Z. Yang, T. Ke, Y. Xin, Large scale evolutionary optimization using cooperative coevolution, Information Sciences 178 (2008) 2985–2999.

[36] M. Zhang, H. Wang, Z. Cui, J. Chen, Hybrid multi-objective cuckoo search with dynamical local search, Memetic Computing (2017) 1–10.

[37] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Transactions on Evolutionary Computation 11 (2007) 712–731.

[38] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (1999) 257–271.