Pattern Recognition Letters 000 (2016) 1–9

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Scaling-up multiobjective evolutionary clustering algorithms using stratification

Alvaro Garcia-Piquer[a], Jaume Bacardit[b], Albert Fornells[c,*], Elisabet Golobardes[d]

[a] Institute of Space Sciences (IEEC–CSIC), Campus UAB, C/ Can Magrans s/n, 08193 Barcelona, Spain
[b] School of Computing Science, Newcastle University, Claremont Tower, Newcastle, NE1 7RU, UK
[c] Research Group in Hospitality, Tourism and Mobilities, HTSI, Universitat Ramon Llull. Av. Marqués de Mulhacén 40–42, 08034 Barcelona, Spain
[d] GR-SETAD, La Salle, Universitat Ramon Llull. Av. Quatre Camins 30, 08022 Barcelona, Spain

ARTICLE INFO

Article history:
Available online xxx

MSC:
41A05
41A10
65D05
65D17

Keywords:
Multiobjective evolutionary algorithms
Clustering
Scaling-Up
Stratification

ABSTRACT

Multiobjective evolutionary clustering algorithms are based on the optimization of several objective functions that guide the search following a cycle based on evolutionary algorithms. Their capabilities allow them to find better solutions than with conventional clustering algorithms when more than one criterion is necessary to obtain understandable patterns from the data. However, these kind of techniques are expensive in terms of computational time and memory usage, and specific strategies are required to ensure their successful scalability when facing large-scale data sets. This work proposes the application of a data subset approach for scaling-up multiobjective clustering algorithms and it also analyzes the impact of three stratification methods. The experiments show that the use of the proposed data subset approach improves the performance of multiobjective evolutionary clustering algorithms without considerably penalizing the accuracy of the final clustering solution.

1. Introduction

Multiobjective clustering (MC) algorithms [19] tackle the challenge of optimizing several criteria that cannot be combined in a single objective function by defining an objective function for each criterion and by optimizing them trying to obtain a trade-off between all the objectives. There are different strategies for multiobjective optimization such as Simulated Annealing [27] and Ant Colony Optimization [23], but Multiobjective Evolutionary Algorithms (MOEAs) [7] have become one of the most capable strategies to solve this kind of problem [29] since they (1) work with a collection of solutions with different trade-offs among objectives, which are improved until a Pareto set with optimal trade-offs is obtained; (2) can be easily adapted to the type of data of the studied domain, due to the flexible knowledge representation used; and (3) are able to optimize different objectives without assuming any underlying structure of the objective functions. Therefore, MOEAs offer outstanding search capabilities but their performance can be compromised in large databases due to their high computational and memory usage requirements [12]. This is an important issue considering the needs when analyzing large data sets in a reasonable computational time and memory usage without considerably penalizing their accuracy [24]. Evolutionary Algorithms (EAs) are based on the principles of evolution and natural selection, applying an iterative process where a collection of initial solutions (individuals) are evolved through pseudo-random recombination until obtaining an optimal solution. In clustering, an individual is a possible group of data. From the whole process, the evaluation is the most time-consuming step because each individual has to be assessed with respect to all the elements according to all the objective functions defined.

This work proposes to scale-up MOEAs when they are applied to large data using an approach based on data subsets techniques. Specifically, the approach splits the data set into several strata so the EA only uses a stratum for evaluating the individuals in each generation following a Round Robin policy to avoid bias problems. Thus, computational and memory costs associated to the evaluation of the population are drastically reduced and its application does not need to modify the algorithm structure. An ideal stratification strategy is to map the initial data set into disjoint strata of equal size and with equal class distribution and where the number of strata is defined by the user. However, clustering problems are unsupervised and classes cannot be used to split the instances into representative strata because they are unknown [3]. Therefore, the definition of the size of subsets and the selection of their elements are not trivial and they influence the performance of the algorithm if they are not sufficiently representative. For this reason,

* Corresponding author.
E-mail addresses: albert.fornells@htsi.url.edu, afornells@gmail.com (A. Fornells).

http://dx.doi.org/10.1016/j.patrec.2016.12.001
0167-8655/

two unsupervised and one supervised strata generation strategies are presented and analyzed: (1) random strata, (2) strata according to clusters distribution using a fast and approximate clustering algorithm and (3) strata based on classes. Moreover, strategies are integrated in a MC algorithm based on MOEAs called CAOS (Clustering Algorithm based on multiObjective Strategies) [15,16] and they are tested with several strata size using artificial and real world problems. Finally, results are compared with each other and with regard to MOCK, which is one of the most well-known MC algorithms based on MOEAs [19]. The results show improvement in computational time while accuracy is not substantially penalized when stratification approaches are applied. Furthermore, the three strategies for building the strata are equivalent so the proposed data subset approach can be used in clustering problems because it does need a stratification method based on classes. Finally, the results also show that the proposed approach is significantly better than MOCK in terms of computational time and accuracy.

The paper is organized as follows. Section 2 summarizes the related work on data subsets applied to clustering.Sections 3 and 4 describe CAOS and the stratification strategies. Section 5 describes the experimentation and discusses the results. Finally, Section 6 ends with conclusions and further work.

2. Related work

Two of the most used strategies for scaling-up EAs are Parallel EA [5] and data subsets [3,10]. The first strategy distributes the computational cost of the evaluation step by parallelizing the evaluation of individuals so it is necessary to adapt or redefine the algorithm in order to be able to parallelize it in a environment with several processors. Moreover, the parallelization may imply an additional communication cost that could decrease the performance achieved with the distribution of compute. On the other hand, the second strategy uses a data subset from the original data set to evaluate the individuals so fewer resources are required and there is no need to modify the algorithm structure. In contrast, the data sets definition is not trivial.

There are two main ways to work with data subsets: using only one of the built data subsets, or using alternatively all the data subsets. The algorithm CLARA (Clustering LARge Applications) [25], one of the most representative algorithms for clustering large data sets, works using the first approach. This algorithm is based on selecting randomly a sample from the entire data set and, subsequently, it finds k medoids of the sample using only the built sample. After this, all the instances of the entire data set are assigned to the most similar medoid. The execution of the entire process is repeated five times, and the solution with less dissimilarity is returned as the solution. Following this idea, other methods consist of randomly extracting several samples from the entire data set and applying the same clustering algorithm to each one of the samples, thereby obtaining several clustering results. After this, all the obtained results are merged in a single clustering solution. Hore et al. [21] proposed using k-means or fuzzy k-means algorithms with large data. The idea is to obtain a set of jointed or disjointed samples and apply one of the two algorithms to each sample to obtain several clustering results. The last step consists of doing a consensus between each clustering result to obtain a final clustering solution as in ensemble clustering. The drawback of using only one sample to obtain the clustering results is that it is necessary to execute the algorithm several times or apply it to different data subsets in order to avoid the bias of using only one sample. Moreover, only a part of the entire data set is used. Thus, the approaches based on using all the data subsets can be useful to obtain the clustering results in a single execution.

ILAS (Incremental Learning by Alternating Strata) [1] is a technique based on Evolutionary Algorithms for classification problems based on dividing the training set into several strata based on using a different stratum in each iteration of the evolutionary algorithm using a round-robin policy. Thus, the individuals are evaluated with all the strata, avoiding any bias of the data and increasing the generalization of the individual. The strategy followed in this paper is based on the ILAS algorithm but applied to MC problems.

3. CAOS

CAOS [15] is a multiobjective evolutionary algorithm system to solve clustering problems based on adapting the multiobjective optimization algorithm PESA-II [8] due to its competitiveness with respect to the state-of-the-art clustering methods and its ability to evolve accurate clusterings from domains with complex structures [19]. It evolves a set of mutually non-dominated clustering solutions (called Pareto set) that correspond to different tradeoffs between objectives. A solution S is non-dominated when there is not any solution better than S in all the objectives. Otherwise, the solution is dominated.

Algorithm 1: Scheme of PESA-II algorithm.

```
1  Let EP and IP be an external and an internal population respectively. They
   store a maximum of N_EP and N_IP individuals, where (N_IP < N_EP)
2  Init. IP with N_IP individuals stochastically created
3  Init. the EP individuals with non-dominated clustering results from IP
4  Evaluate all the individuals from EP according to the objectives
5  foreach Generation do
6      Select N_IP individuals from EP to generate a new IP
7      while (|IP| != ∅) do
8          Select and remove two individuals from IP
9          Cross and mutate them to obtain 2 new ind.: I_New1 and I_New2
10         foreach I_Newi do
11             Evaluate the I_Newi fitness according to the objectives
12             if I_Newi dominates any individual from EP then
13                 Remove the dominated individuals by I_Newi from EP
14                 Add I_Newi into EP;
15             else if I_Newi is not-dominated and I_Newi not-dominates any
               individual then
16                 if EP is full then
17                     Remove an ind. from the most crowded niche
18                 Add I_Newi into EP
19 Select a individual from EP as a solution
```

Algorithm 1 summarizes the main elements of PESA-II. It evolves an external population (*EP*) of individuals through a number of generation where individuals are selected, crossed and mutated following the typical evolutionary cycle. Individual are represented with real numbers that represent the coordinates (attributes) of the cluster prototype using a centroid-based representation [22]. More specifically, each individual consists of $n \cdot t$ genes $\{x_{11}, \ldots, x_{1t}, \ldots, x_{n1}, \ldots, x_{nt}\}$, where n is the number of clusters of the individual, t is the number of the attributes of the data set, and x_{ij} is the value of the attribute j of the cluster centroid i. The genotypic representation is transformed into the phenotypic representation by assigning each instance to the cluster with the nearest centroid to it. In addition to EP, it also maintains an *internal population* (*IP*) to separate the exploration from the storage of the best solutions. That is, *IP* is used to explore new promising solutions and *EP* is employed to store a large and diverse set of non dominated solutions found so far. Moreover, *EP* is organized in N_{niches} different niches through the placement of an hyper-grid in the objective space splitting it in hyper-rectangles, where each of them is considered as a separate niche. Therefore, solutions with similar objectives will be placed in the same niche. The replacement process uses the niching mechanism to make