REVISTA DE MATEMÁTICA: TEORÍA Y APLICACIONES 2016 **23**(2) : 445–461
CIMPA – UCR ISSN: 1409-2433 (PRINT), 2215-3373 (ONLINE)

CLUSTERING PROBLEMS IN A MULTIOBJECTIVE FRAMEWORK

PROBLEMAS DE AGRUPACIÓN EN UN AMBIENTE MULTIOBJETIVO

YUNAY HERNÁNDEZ* RICARDO BEAUSOLEIL†

Received: 14 Oct 2014; Revised: 09 Sep 2015;
Accepted: 20 Feb 2016

*Department of Interdisciplinary Mathematics, Institute of Cybernetics, Mathematics and Physics, Havana, Cuba. E-Mail: yunay@icimaf.cu

†Misma dirección que/*Same address as*: Y. Hernández. E-Mail: rbeau3105@gmail.com, rbeausol@icimaf.cu

Abstract

We propose a new algorithm using tabu search to deal with bi-objective clustering problems. A cluster is a collection of records that are similar to one other and dissimilar to records in other clusters. Clustering has applications in VLSI design, protein-protein interaction networks, data mining and many others areas. Clustering problems have been subject of numerous studies; however, most of the work has focused on single-objective problems. In the context of multiobjective optimization our aim is to find a good approximation to the Pareto front and provide a method to make decisions. As an application problem we present the zoning problem by allowing the optimization of two objectives.

Keywords: combinatorial data analysis; clustering; tabu search; multiobjective optimization.

Resumen

En este trabajo proponemos un nuevo algoritmo usando un enfoque de búsqueda tabú para dar solución a problemas de agrupación (clusters) tomando en consideración dos objetivos. La tarea de agrupación se refiere a la agrupación de objetos, observaciones, o casos. Una agrupación es una colección de objetos similares entre sí y disímiles entre agrupaciones. Aplicaciones de agrupaciones tienen lugar en los diseños VLSI, redes de interacción proteína-proteína, minería de datos y muchas otras áreas. Los problemas de agrupación han sido ampliamente estudiados, pero su descripción se ha basado en la consideración de solamente un objetivo. En el contexto de optimización multiobjetivo nuestro objetivo es hallar una buena aproximación de la frontera Pareto y proveer un método para la toma de decisión. Como aplicación presentamos el problema de zonificación optimizando dos objetivos.

Palabras clave: Análisis de datos combinatorio; cluster; búsqueda tabú; optimización multiobjetivo.

Mathematics Subject Classification: 90C27, 90C29, 90C30, 90B50, 93B40.

1 Introduction

Cluster analysis divides data into groups that are meaningful, useful, or both. If meaningful groups are the goal, then the groups (clusters) should capture the natural structure of the data. Cluster analysis has long played an important role in a wide variety of fields: psychology and other social sciences, biology, statistic, pattern recognition, information retrieval, machine learning, and data mining.

Clustering deals with problems of classification of a set of objects that share common characteristics. The goal is that the objects within a group be similar to one another and different from the objects in other groups. Different types of clustering exist, traditionally, two main approaches: hierarchical clustering methods and partition clustering methods [11], [12]. Hierarchical approach, essentially heuristic procedures, produce a hierarchy of partitions of the set of objects according to an agglomerative strategy or to a divisive one. Partition approach, in general, assume a given number of clusters and seek the optimization of an objective function measuring the homogeneity within the clusters and/or the separation between the clusters. Heuristic algorithm, as the traditional k-means algorithm [14] is frequently used to find a local minimum of the objective function. However, any mathematical programming technique can be apply to solve the global optimization problem, in particular, all kinds of metaheuristics [17], [15].

Caballero et al. [6] consider two type of multiobjective partitioning problems: 1) partitioning of objects using a simple objective function with multiple dissimilarity matrices, and 2) generating classes using several objective functions. Handl and Knowels use two competitive objective functions and apply a multiobjective evolutionary algorithm [10]. Brusco and Stahl describe the case of a telecommunication company that would like of business customers based on the demographics and also based on current satisfaction and the willingness to switch providers [5]. Brusco and Cradit create bipartite models [4]. As in Bernabe et al. [3] we describe and solve the zoning problem with two objectives. The compactness meaning the cluster given to the set of objects (territorial units for a zoning problem) represents one of those objective to be optimize, the other, homogeneity meaning a balance in the distribution of several variables that each object in the problem possesses and are selected as input for the algorithm, both objectives are being minimize.

2 Basic concepts

A general multiple objective optimization (MOO) problem consists of optimizing a set of $r \geq 2$ objective functions. It can be formulated as follows:

$$\begin{array}{ll} \text{minimize} & f(s) : f(s) = (f_1(s), f_2(s), \ldots, f_r(s)) \\ \text{s.t.} & s \in X \end{array}$$

where a solution $s = (s_1, s_2, \ldots, s_n) \in X$ is represented by a vector of n decision variables, X is a set of feasible solutions. The image of a solution s in the objective space is a point $z = (z_1, z_2, \ldots, z_r) = f(s)$.

Having several objectives functions, the notion of optimum changes. The aim here is to find a good compromise rather than a unique solution as in a mono-objective optimization problem. A MOO problem obtains rather a set of solutions known as the Pareto optimal, related to the following concepts:

Definition 1 (Pareto dominance) *A solution $s^1 \in X$ dominates another solution $s^2 \in X$ if and only if $\forall i \in \{1, 2, \ldots, n\}$, $f_i(s^1) \leq f_i(s^2)$, and $\exists j \in \{1, 2, \ldots, n\} : f_j(s^1) < f_j(s^2)$.*

Definition 2 (Efficiency) *A solution s^* is efficient if and only if there is not another solution $s \in X$ such that s dominates s^*.*

The whole set of efficient solutions is the Pareto optimal set, and is denoted by X_P. The image of a Pareto optimal set in the objective space results in a set of non-dominated vector denoted by PF and called *non-dominated set* or Pareto Frontier.

The aim in multiobjective metaheuristic optimization is to obtain a Pareto optimal set or a good approximation to it.

3 Multiple criteria and dissimilarity matrixes

In this work, as in Caballero et al. [6], we consider two type of multiobjective partitioning problems: 1) partitioning of objects using a simple objective function with multiple dissimilarity matrixes, and 2) generating classes using several objective functions. We use here objective functions considered in the work of Brusco and Stahl [5] and also in [6] that consist of minimizing the combination of the following functions: Partition diameter (D), Unadjusted within-cluster dissimilarity (UWC), Adjusted within-cluster dissimilarity (AWC), Average within-cluster dissimilarity (AvWC), where

$$D = \max_{k=1..K} \Big(\max_{(i<j) \in C_k} (a_{ij}) \Big) \tag{1}$$

$$UWC = \sum_{k=1..K} \sum_{(i<j) \in C_k} a_{ij} \tag{2}$$

$$AWC = \sum_{k=1..K} \sum_{(i<j) \in C_k} a_{ij}/n_k \tag{3}$$

$$AvWC = \sum_{k=1..K} \sum_{(i<j) \in C_k} a_{ij}/(n_k(n_k-1)/2) \tag{4}$$

with C_k means the k cluster, and $\|a_{ij}\|$ the matrix of dissimilarity.

However, there are situations where the partitions are more effective if several data sources are used. These sources can be aggregated to perform a traditional cluster analysis, but sometimes it is more interesting to search compactness for some variables on one source of data, while identifying the homogeneity of the variables on the second source of data [3]. In this case we study the following problem: the Optimal Zoning, that consist in to obtain a spatial data partitioning named BGAs (Basic Geostatistical Areas). Its composition consists of two components: geographical coordinates in the plane $\mathbb{R}^2$ and a vector of census descriptive characteristics. From this point of view, we look for a partition conformed by a set of classes with components that are very close geographically, and balanced according of its census variables. To do this, we use an objective function that minimizes the sum of the distances between the elements of the BGAs of each group and its center. The homogeneity is optimized seeking a grouping balance in some census variables of interest.

$$\text{Compactness} = Minimizer\{\sum_{k=1..K}\sum_{j,c\in C_k} a_{jc}\} \tag{5}$$

$$\text{Homogeneity} = Minimizer\{\sum_{k=1..K}\sum_{j\in C_k}\sum_{i=1,\ldots,n'} |\overline{x}_i - x_{ij}|\} \tag{6}$$

where $c = center$, n' : number of demographic variables selected for analysis, $\overline{x}_i = \sum_{j=1..m} x_i^j/m, \forall i = 1,\ldots,n', m$: number of object that has the database.

4 Ranking the set of non-dominated solutions

Clustering algorithms seek to construct clusters of records such that the between-cluster variation (BCV) is large compared to the within-cluster variation (WCV). Using $\min\{a_{c\overline{c}}\}\ \forall(c,\overline{c})$with c and $\overline{c}$ centers as a surrogate for BCV and $\max\{\max\{a_{i,c}\}\}\ \forall(i,c)$ with $i,c \in C_j, j \in \{1,\ldots,K\}$ as a surrogate for WCV, we have:

$$\frac{BCV}{WCV}. \tag{7}$$

This indicator can be used to rank the set of non-dominated clusters obtained as results of the multiobjective search. The best solution will be the that possesses higher value obtained by the ratio that shown in (7).

5 Metaheuristic search: A tabu framework

Here, we present an adaptation of our multiple criteria scatter search to deal with multiobjective clustering [2]. The details of our approach is presented below.

Tabu Search (TS) was proposed in its present form by Glover [9]. TS can be described as an intelligent search that use adaptive memory and responsive exploration. It is an iterative technique which explore a set of problem solutions, denoted by X, by repeatedly making moves from one solution s to another solution s' located in the neighborhood $N(s)$ of s.

5.1 Neighborhood

To explain the method to generate neighbors, first we explain how the solution is encoded. The solution s is a permutation of n integer number, for example, $s = (1, 3, 4, 6, 8, 7, 9, 10)$ where the first k numbers represent the centers of the k clusters and the rest of the numbers represent the objects to set into the clusters. Then, assume that $k = 3$ a neighbor of the solution s is $s' = (1, 9, 4, 6, 8, 7, 3, 10)$, that is, we interchange the pair $(3, 9)$. This representation is also used in [6].

Two methods of generating neighbor are presented in our approach; the first method consist of every possible pairwise interchange operation, where the first component of the pair is a current center and the second one object to set into one cluster; the second method consist of all two pairwise interchange, that is, to change two center in order to diversify the search.

5.2 Memory structures

The main components of tabu search approaches are memory structures, in order to have a trace of the evolution of the search. TS maintains a selective history of the states encountered during the search, and replaces $N(s)$ by a modified neighborhood $N^*(s)$. In our implementation we use recency and frequency memories. Associate with these memories exist a memory structure, so called tabu list that takes account the identity of the pairs of elements that changed positions. In order to maintain the tabu list, one array *tabuend(e)* is used, where *e* ranges the attributes.

A complementary memory structure associated with the array *tabuend(e)* we have, the array *freq_count* showing the frequency distribution of the centers, that is, the number of time that each center takes part in the visited solutions. Our algorithm uses this frequency information to penalize moves that not qualify as efficient move and to favor moves that qualify as efficient move.

5.3 Moves and tabu restrictions

In our implementation we propose to choose moves that change one or two centers by one or two objects that not qualify as current centers, that is, pairs (i, j) where i is a current center of one cluster and j is a current object to set into this cluster.

Recorded move attributes are often used in tabu search to impose constraints, that prevent moves from being chosen that would reverse the changes represented by these attributes. In our case, we impose tabu restriction to the center (or centers) that was (or were) changed. Initially, *tabuend(i)* is zero and it is incremented in the number of time that this attribute e has been used as center when a move from s^{now} (s^{now} is the current solution) to s^{next} is performed. This operation prevents a move that contain it as a new center during the time given by *tabuend(i)*.

5.4 Search by goals

Let S the set of trial solutions. A thresholding aspiration is used to obtain an initial set of solutions as follows: without lost generality, let us assume that every criteria is minimized. Notationally, let $\Delta f(s^{'}) = (\Delta f_1(s^{'}), \ldots, \Delta f_r(s^{'}))$ where $\Delta f_k(s^{'}) = f_k(s^{'}) - z_k^*$, $k \in \{1, \ldots, r\}$.

A goal is satisfied, permitting $s^{'}$ to be accepted and introduced in S if $(\exists \Delta f_k(s^{'}) \geq 0)$ *or* $(\forall k \in \{1, \ldots, r\}[\Delta f_k(s^{'}) = 0])$, otherwise is rejected.
The point Z^* is updated by $z_k^* = \max f_k(s^{'}) \forall k \in \{1, \ldots, r\}$, $s^{'} \in S$.

In order to measure the quality of the solution we propose to use in our tabu search approach an additive function afv with weighting coefficients λ_k ($\lambda_k \geq 0$), representing the relative importance of the objectives. We want to set the weights λ_k ($k = 1..r$) so that the solution selected is closest to the new aspiration threshold. Therefore each component in the weight vector is set according to the objective function values. We would give more importance to those objectives that have greater differences between the quality of the trial solution and the quality of the reference solution. The influence is given by an exponential function $\exp(-\tilde{s_k})$, where $\tilde{s_k}$ is obtained as $\tilde{s_k} = \Delta f_k(s^{'})/z_k^*$, $\lambda_k = 2 - \exp(-\tilde{s_k})$ $(k \in \{1, \ldots, r\})$, then $afv(s^{'}) = \sum_{k=1,r} \lambda_k \Delta f_k(s^{'})$.

5.5 Diversification and intensification strategies

The diversification stage encourages to generate solutions that differ in various significant ways from those seen before. Intensification strategies in generate solutions that has been found good. In order to implement these strategies, we use the following strategies: 1) diversification: a large step move is executed, that consist in select two center and to change these by two objects with low frequency *freq_count*, this information is also used to penalize non-efficient move as follow: let $v = afv(s^{now}) - afv(s^{next})$ and sum_freq the addition of all objects that has played the roll of center, then the penalize function is formulated as $P = v * (1 + freq\_count(p) + freq\_count(q)/sum\_freq)$. 2) intensification: a short step move is executed, that consist in select one center and to change by one object free to choose; the frequency information is used here to induce to select efficient move by the following function $I = v * (1 - freq\_count(p)/sum\_freq)$. In the two cases we assume that $sum\_freq \neq 0$.

5.6 The reference set

Let $R \subseteq S - C$ the set of current non-dominated solutions that we call the reference set and C the set of critical solutions, consisting of duplicated solution. We consider a duplicated solution if the following conditions holds: $f_1(s) = f_1(s^{'})$ and$f_2(s) = f_2(s^{'})$ and $size\_c(s) = size\_c(s^{'})$ where $s \in R$, $s^{'} \in S$ and $size\_c(s)$ is the cardinal of the cluster associated to the solution s.

5.7 Initialization

In the initialization phase a consecutive sequence of n natural numbers, as labels of the objects, are generated. The first number is chosen as center of the first cluster. The second center is the element with the maximum Euclidean distance. Now we have two center, the following centers are formed chosen the elements with the maximum distance that is greater than the average distance of the chosen centers. If the above condition is not satisfied, then the element with nearest distance to the average is chosen.

5.8 Restart points

In order to resume the process of search, we take starting points from R. Set $startingpoints = p \in R$ where p is chosen taking into account that it has not been explored.

6 Algorithm

1. Generate an initial solution s^{now}.
2. Set $f(s^{now})$ as reference point.

3. Generate a neighborhood $N^*(s^{now})$.

4. If there does not exist new efficient move then, increase the counter of non new efficient move, and if the number of moves reaches a fixed threshold then initialize a diversification phase; otherwise, reset the counter of non new efficient move and go out of the diversification phase.

5. Repeat from 3. until a cut-of-rule is reached.

6. Update the set of reference points with news non-dominated points.

7. Chose a new non-dominated point to restart the search and reset all memories.

8. Repeat from 2. until a maximum number of iteration is reaches or does not exits news non-dominated points.

7 Data structure

There are two different types of input data: vector sets and dissimilarity matrixes. The first kind of input data is a set of *n*, *k*-dimensional vectors. The dissimilarity between vectors is generated calculating the Euclidean distance between the corresponding pair of vectors. The second kind of input data are the dissimilarity matrixes. It is a symmetric matrix, for this reason only is necessary recorded a triangular matrix.

8 Computational simulation

We perform experiments with a well known benchmark from UCI Machine Learning and Intelligent Systems Database.

8.1 Vector sets

A vector set is a collection of *n*, *k*-dimensional real vectors. The vector sets that we used in our experiments are: the Wine data set, Glass data set and Iris data set:

Wine data set: These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines (see [1]). The data set contains 178 objects distributed in three classes: class 1 = 59, class 2 = 71 and class 3 = 48.

Glass data set: The Glass data set describes types of glass. This was motivated by criminological research. The testing database has 214 objects with 10 attributes for each patient. The class distribution is: class 1 = 87 objects, class 2 = 76.

Iris data set: The Iris data set by Fisher [8] is the most famous data set used, contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2, the latter are not linearly separable from each other.

Table 1 shows the 3-model $D, AWC, AvWC$ for the Iris problem. In this, one cluster with the best ranking presents 50, 45 and 55 objects in each cluster. The first column shows the data sets used. The second column the number of clusters, the following columns the objective functions (1), (3) and (4). The last column shows the ranking of each solution achieved, and the black row is the best solution of the Pareto set according to the measure defined in the section 4.

Table 1: Performance for 3 objectives and 3 clusters.

Database	D	AWC	AvWC	BCV/WCV
Iris	**6.717**	**153.221**	**6.219**	**0.508**
	6.717	153.168	6.251	0.078
	6.717	154.865	6.206	0.273
	6.717	155.047	6.197	0.304
	6.717	155.494	6.143	0.304
	6.869	156.298	6.125	0.297
	6.869	157.657	6.074	0.158
	6.869	157.650	6.120	0.206
	6.717	158.984	6.129	0.304
	6.869	158.959	6.062	0.297
	6.791	160.308	6.114	0.300
	8.094	166.486	5.979	0.248
	9.821	182.946	5.552	0.208
	9.821	182.946	5.552	0.208
	9.821	183.206	5.497	0.208

8.2 Validation of the results

Below is the validation algorithm presented using the databases showed. Measures to evaluate the results are: the Rand index [16], [18] and the Jaccard index [18]. These external indexes are, in general, used to measure similarity between two sets elements:

$$RandIndex = \frac{a+d}{M}$$

$$JaccardIndex = \frac{a}{a+b+c}.$$

Given C as the true partition, P the cluster reached by the algorithm and (x_i, x_j) pair of elements S, with M cardinal of S, we have:

- a, the number of pairs of elements (x_i, x_j) in S that are in the same set in C and in the same set in P,
- b, the number of pairs of elements (x_i, x_j) in S that are in the same set in C and in different sets in P,
- c, the number of pairs of elements (x_i, x_j) in S that are in different sets in C and in the same sets in P,
- d, the number of pairs of elements (x_i, x_j) in S that are in different sets in C and in different sets in P.

Intuitively, $a + d$ can be considered as the number of agreements between C and P and $b + c$ as the number of disagreements between C and P. Terms a and d are measures of consistent classifications (agreements), whereas terms b and c are measures of inconsistent classifications (disagreements). Note that for the Rand index: 1) the result belongs to $[0, 1]$, 2) if the Rand index is cero then P is completely inconsistent, and if the Rand index is one then P is completely consistent with C, i.e. $P \equiv C$.

The Jaccard index belongs to $[0, 1]$, viewed as counts of “good pairs" with of term a and “bad pairs" with of terms b and c. From this viewpoint, the Jaccard index can be seen as a proportion of good pairs with respect to the sum of non-neutral (good plus bad) pairs, whereas the Rand index is just the proportion of pairs not definitely bad with respect to the cardinal of the data base.

Tables 2, 3, and 4 show the results achieved for the data sets: Iris, Wine and Glass respectively. In this tables, the first column shows the data sets used, the second the rand index and the following column the Jaccard index. In table 2 the rand index shows that 72.07 to 89.86 percent of the objects are correctly

Table 2: Validation for the Iris data.

Database	Rand index	Jaccard index
Iris	0.8956	0.7294
	0.8829	0.7032
	0.8991	0.7347
	0.8925	0.7206
	0.8682	0.6720
	0.8709	0.6799
	0.8490	0.6414
	0.8542	0.6501
	0.8325	0.6122
	0.8345	0.6188
	0.8301	0.6124
	0.8053	0.5875
	0.7207	0.4946
	0.7207	0.4946
	0.7236	0.4998
	0.7236	0.4998
	0.7261	0.5043
	0.7727	0.5784
	0.7300	0.5086

assigned, and the Jaccard index shows that 49.46 to 72.94 percent of the objects are correctly assigned.

In table 3 the rand index shows that 87.42 percent of the objects are correctly assigned, and the Jaccard index shows that 65.18 percent of the objects are correctly assigned. In table 4 the rand index shows that 66.47 to 70.43 percent of the objects are correctly assigned, and Jaccard index shows that 19.72 to 24.86 percent of the objects are correctly assigned.

Table 3: Validation for the Wine data.

Database	Rand index	Jaccard index
Wine	0.8742	0.6518

Table 4: Validation for the Glass data.

Database	Rand index	Jaccard index
Glass	0.6979	0.2455
	0.6995	0.2480
	0.7001	0.2447
	0.7043	0.2486
	0.7021	0.2486
	0.7038	0.2450
	0.6986	0.2351
	0.6844	0.2249
	0.6754	0.2341
	0.6945	0.2130
	0.6891	0.2117
	0.6957	0.2294
	0.6793	0.2341
	0.6782	0.2351
	0.6795	0.2359
	0.6891	0.1938
	0.6723	0.2315
	0.6727	0.2325
	0.6827	0.2308
	0.6916	0.1972
	0.6767	0.2380
	0.6687	0.2271
	0.6758	0.2373
	0.6748	0.2383
	0.6883	0.2307
	0.6713	0.2295
	0.6647	0.2244

8.3 Dissimilarity matrixes

In this problem we consider more than one dissimilarity matrix. There are some cases where is more appropriate to formulate the problem as a multiobjective partitioning of objectives using multiple dissimilarity matrixes. Here we present the zoning problem that can be defined as a grouping process of geographical areas; it appeared for the first time to separate the residential areas of the industrial ones. Here, the main issue is to know how some census variable are distributed or concentrated in certain territorial spaces. To solve this problem we work with two dissimilarity matrixes, that can represent the distances between areas and the census variables.

The following example was taken from [3]: the metropolitan area of Toluca Valley is going to be grouped in compact and homogeneous partitions that only include elements whose variables have values in the ranges indicated below. It is important to note that these variables are bounded in a value that is above the average:

- Male Population under 6 years.
- Male population between 6 and 11 years.
- Male population between 15 and 17.
- The homogeneity will be obtained on the variable.

Table 5 shows the results for three and five clusters and the figure 1 shows the front for the five clusters. In this case the results show that the best is considering three clusters to according the objectives functions (5) and (6).

9 Conclusion and future works

We have described the development and testing of a metaheuristic approach in multiobjective clustering problems. The proposed procedure is based on a well known metaheuristic methodology, tabu search. We show that our procedure obtains good approximations by applying and comparing it to several data test sets for clustering problems with several objective functions and also with different dissimilarity matrices. The results show that our procedure is very fast, obtaining approximation to Pareto fronts in all cases in less than 30 seconds.

Our method can also be applied to other problems adjusting the implementation of the particular problem.

Table 5: Performance for clusters with 2-dissimilarity matrixes.

Zonification	Com	Hom	BCV/WCV
3-clusters	58.775	8366.817	0.044
	68.461	8211.841	0.041
	65.873	8305.590	0.026
	57.734	8403.814	0.040
	55.803	8496.392	0.027
5-clusters	73.113	9087.793	0.041
	77.429	8978.929	0.012
	71.867	9099.908	0.033
	71.735	9239.288	0.017
	69.822	9273.665	0.027
	69.179	10242.812	0.029
	68.475	10554.774	0.020
	62.922	11807.978	0.020

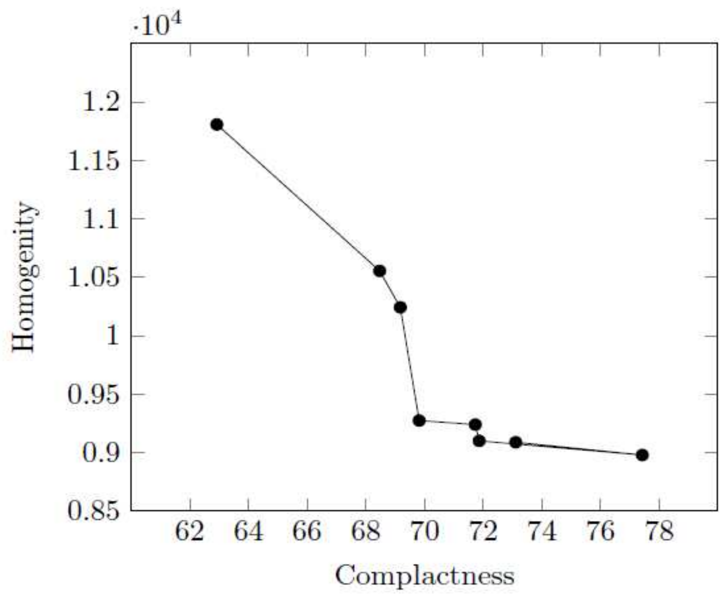

Figure 1: Non-dominated solutions on the Zoning problem considering five clusters.

Acknowledgements

The authors acknowledge B. Bernábe-Loranca for providing the data set in the zoning problem.

References

[1] Aeberhard, S.; Coomans, D.; de Vel, O. (1994) "Comparative analysis of statistical pattern recognition methods in high dimensional settings", *Pattern Recognition* **27**(8): 1065–1077.

[2] Beausoleil, R.P. (2001) "Multiple criteria scatter search", in: *Proceedings MIC'2001- 4th Metaheuristics International Conference*, Porto, Portugal: 539–543.

[3] Bernábe, B.; Coello, C.A., Osorio, M. (2012) "A multiobjective approach for the heuristic optimization of compactness and homogeneity in the optimal zoning", *Journal of Applied Research and Technology* **10**(3): 447–457.

[4] Brusco, M.J.; Cradit, J.D. (2005) "Bicriterion methods for partitioning dissimilarity matrices", *British Journal of Mathematical and Statistical Psychology* **58**(2): 319–332.

[5] Brusco, M.J.; Stahl, S. (2005) *Branch-and-Bound Applications in Combinatorial Data Analysis*. Springer, New York.

[6] Caballero, R.; Laguna, M.; Martí, R.; Molina, J. (2006) "Multiobjective clustering with metaheuristic optimization technology", *Technical Report 02-06, Leeds School of Business in the University of Colorado at Boulder, CO*, available in: `http://www.uv.es/sestio/TechRep/tr02-06.pdf`

[7] Delattre, M.; Hansen, P, (1980) "Bicriterion cluster analysis", *IEEE Transaction on Pattern Analysis and Machine Intelligence* **2**(4): 227–291.

[8] Fisher, R.A. (1936) "The use of multiple measurements in taxonomic problems", *Annals of Eugenics* **7**(2): 179–188.

[9] Glover, F. (1986) "Future paths for integer programming and links to artificial intelligence", *Computers and Ops. Res.* **13**(5): 533–549.

[10] Handl, J.; Knowles, J. (2004) "Evolutionary multiobjective clustering", in: X. Yao et al. (Eds.) *Parallel Problem Solving from Nature (PPSN VIII)*, Lecture Notes on Computer Science 3242, Springer, Heidelberg: 1081–1091.

[11] Hansen, P.; Jaumard, B. (1997) "Cluster analysis and mathematical programming", *Mathematical Programming* **79**(1-3): 191–215.

[12] Jain, A.K.; Dubes, R.C. (1988) *Algorithms for Clustering Data*. Prentice-Hall, Upper Saddle River NJ.

[13] Kurgan, L.A.; Cios, K.J.; Tadeusiewicz, R.; Ogiela, M.; Goodenday, L.S. (2001) "Knowledge discovery approach to automated cardiac SPECT diagnosis", *Artificial Intelligence in Medicine* **23**(2): 149–169.

[14] Mc Queen, J. (1967) "Some methods for classification and analysis of multivariate observations", in: *Proceedings of the Fifth Berkeley Symposiums on Mathematical Statistics and Probability, Vol. 1*: 281–297.

[15] Pacheco, J.; Valencia, O. (2003) "Design of hybrids for the minimum sum-of-squares clustering problem", *Computational Statistics and Data Analysis* **43**(2): 235–248.

[16] Rand, W.M. (1971) "Objective criteria for the evaluation of clustering methods", *J. Am. Stat. Assoc.* **66**(336): 846–850.

[17] Reeves, C.R. (Ed.) (1993) *Modern Heuristics Techniques for Combinatorial Problems*. Wiley, New York.

[18] Vendramin, L.; Campello, R.J.G.B.; Hruschka, E.R. (2010) "Relative clustering validity criteria: A comparative overview", *Statistical Analysis and Data Mining* **3**(4): 209–235.