Tendências em Matemática Aplicada e Computacional, **1**, No. 2 (2000), 303-317.
Sociedade Brasileira de Matemática Aplicada e Computacional.

A Recovering Comparative Study of Clustering Analysis

M.A. BIAGIO[1], Statistic Department, University of Brasilia, Brazil.

Abstract. Metaheuristic methodologies have been recently proposed for the minimum sum-of-squares clustering (MSSC) problem and tested on some standart problems from the literature, showing they have promising computational results. However it is not known their recovery performances for data samples randomly generated following a multivariate mormal mixture distribution and with well-separated cluster structures. By generating these kind of samples and computing the Hubert and Arabie recovery measurement, this work presents a performance avaluation of two different versions of the Variable Neighborhood Search metaheuristic (VNS), a multistart k-Means, the hierarchic agglomerative Group Average and two hierarchic divisive clustering algorithms. The results indicate the mean recovery value obtained from the metaheuristics, including multistart k-Means, are in the most of cases better than those ones presented by the hierarchical algorithms. By considering its computational simplicity, the multistart k-Means appears to be a good option for solving MSSC problems, but losing this position to VNS by a litle difference when increasing the number of clusters.

1. Introduction

Cluster analysis is a general statistical methodology for partitioning a set of entities. Its goal is often to develop a classification of the entities which normally refers to their assignment to predefined classes. In this way is useful to emphasize that cluster analysis preceeds classification in the analysis of a data set by requiring, first of all, the identification of the classes. Particularly, the objective of cluster analysis is to assign n entities to k mutually exclusive groups while minimizing some measure of dissimilarity, or criteria, among the entities. In most instances, the measure of dissimilarity is a distance defined by a metric.

Cluster analysis is now used in a wide range of disciplines including archaeology, astronomy, computer science, geology, market research, medicine and psychiatry. Despite the ancient age of the problem of classification, it is known that sophisticated methodologies for clustering have only been proposed after the "boom" of robust computers. This fact is understandable given that clustering problems present combinatorial nature and, consequently, they demand heavy number of operations to be solved. Among the earliest methodologies it should be cited those

[1]mamelia@unb.br

methods of classical statistical cluster analysis of Fisher(1958) and Ward (1963) and of Lance and Williams (1967). Nowaday, the number of methods for clustering and subsequent classification abound in the literature; reviews of them may be found in Gordon (1981), Hand (1981), Hartigan (1975), Anderberg (1973), Everitt (1993) and Mirkin (1996). Discussions of clustering applications and alternative clustering criteria can be found in Anderberg (1973), Sokal and Sneath (1973) and Mac Queen (1979).

The basic premise of the clustering methodologies is to utilize a distance or dissimilarity matrix to group entities together based upon attributes or characteristics. The general methodology requires the determination of a distance matrix and the clustering of entities one by one in a bottom-up approach or decomposing the entire set of items into two groups successively in a top-down approach. These are the strategies used by the well-known class of hierarchical methodologies.

Other well-known classes of methodologies for clustering are the partitioning and heuristics ones. By using different strategies for solving problems, they have recently been receiving special research attention for their characteristic in spending less computer memory than the hierarchic methodologies. Particularly, heuristic methodologies have been intensively applied to acelerate algorithms recently proposed for clustering [19, 11]. In these works, some computational results were presented showing the proposed methodologies are very promising. However, it is well-known from the clustering literature, methodologies have been presenting different performances depending upon the shape of the cluster structures. Metaheuristics recently proposed by N. Mladenovic and P. Hansen have been tested for clustering problems that are described at the end of the next section; however, they do not yet have been their performances compared neither to classical methodologies as hierarchical divisive nor agglomerative ones.

This work presents a comparative computational study which reports the recovery performances of those recent proposed metaheuristics, a multistart k-Means, two hierarchical divisive and the agglomerative Group Average methodologies [1]. The clustering test problems are generated by a programming proposed by G.W. Milligan [17] considering samples that combine both properties of external isolation and internal cohesion of a multivariate normal mixture model with a free-error condition. The results are based on the external criterion of Hubert and Arabic and, separately, of Rand for clustering. Next section gives a view of the state of the art of some clustering methodologies, section 3 describes briefly the VNS metaheuristic, the heuristic H-Means, and the new heuristic J-Means, and also the hierarchic divisive methods by P.Hansen et al [9]. Section 4 shows and analyses the obtained results and section 5 presents conclusions and discussions about the results presented and analysed in the earlier section.

2. State of the Art

2.1. Hierarchical Algorithms

Among the oldest and most used methods of cluster analysis are the agglomerative hierarchical ones. As general methodologies, they follow the first strategy described above for hierarchic classes; the most famous is the Single Linkage (or nearest neighbor) method in which clusters are formed on the basis of the similarity between the most similar pair of entities, one of which is in each pair of clusters. The clusters formed by this methodology have the property that any cluster member is more similar to at least one member of that cluster than it is to any member of any other cluster. Single Linkage is just one of a group of hierarchic agglomerative clustering methods that have been extensively used over the years and that include the Complete Linkage, the Group Average and Ward methods. All of these methods differ just in the definition of similarity that is used for the selection of the most similar pairs of clusters.

The clustering literature seems to indicate that the rank order performance of these methods is not the same from one defining metric to another and from one conceptualization of cluster structure to another. In their work, Milligan et al [16] presented a study using a number of different structures to generate artificial data with all cluster structures involving a total of 24 points. In this simulation study, the four hierarchic agglomerative clustering methods cited above were examined in their ability to recover the true structure in data sets which satisfied both the ultrametric inequality and the structural model of the clustering procedures. The Group Average method had the highest recovery rate and it was followed by the Complete Linkage method. The Ward's method was placed third and the Single Linkage was the last. One possible explanation for the Ward's method performance is that the ultrametric clusters generated in the condition were not necessarily minimum variance; thus, the minimum variance criterion as used by the algorithm was inappropriate for the data. The results indicated that the obtained rank order performance of the methods differed markedly from the rank order generally found in multivariate normal mixture model studies.

Agglomerative hierarchical clustering algorithms optimize a criterion locally at each iteration when two clusters are merged. For a large variety of criteria, agglomerative schemes are polynomial of low-order. The classical agglomerative hierarchical clustering scheme of Lance and Williams (1967) is in $O(n^3)$. This complexity can be reduced to $O(n^2 \times logn)$ by using priority queues as shown by Day and Edelsbrunner (1984) [20].

Other well-known, but less frequently used, general methodologies are the divisive hierarchical clustering algorithms. They follow the second strategy described in the earlier section for hierarchic classes: initially there is one cluster containing all n objects; at each stage in the algorithm an existing cluster is divided into two. Algorithms for dividing a cluster into two involve successively removing entities from the cluster, or selecting the pair of entities in the cluster with largest pairwise dissimilarity to act as the seeds for the two sub-clusters. Algorithms that globally find the optimal division are computationally very demanding. Divisive hierarchical clustering algorithms present a NP-hard local optimization problem of bipartitioning when the variance or the inter-cluster sum-of-dissimilarities criteria are used; but this computational load is significantly reduced for the diameter criterion or for the sum-of-diameters criterion [6, 7, 21].

Few researchers have been proving the superiority of some divisive hierarchic clustering algorithms over agglomerative ones, mainly when the goal is to reach partitions of the entities into a few clusters. Experiments with large randomly generated data sets reported by Guenoche (1991) tend to confirm this superiority for the diameter criterion. The explanations for these results are that, as in hierarchical schemes errors are never corrected, it may be better to obtain partitions in a few steps than in many; moreover, divisive hierarchical schemes with exact bipartitioning procedures of course provide optimal partitions into two clusters. In this same work the authors compare the results of both divisive and agglomerative hierarchical schemes in terms of the partition diameters; among the tested problems, they used a larger one with number of entities varying in a range between 200 and 500 and integer dissimilarities randomly generated in a uniform distribution on [1;100.000]. The proposed algorithms are compared to the agglomerative hierarchical clustering algorithms of Benzecri(1982) and Juan (1982). While the latter is often less time-consuming than the former one, it provides many partitions with larger diameter than those they give. The authors concluded that the proposed divisive hierarchical clustering algorithms appear to be as an alternative when solving minimum diameter clustering problems. Also P.Hansen et al (1993) reported positive computational results in favour to some hierarchical divisive methodologies against agglomerative ones. In their work an exact algorithm and a heuristic variant are proposed for the divisive Average-Linkage hierarchical method. Problems with up to 500 entities are solved approximately in resonable computing time. Comparing partitions obtained by the heuristic variant and those obtained by the agglomerative Average-Linkage hierarchical clustering by Benzecri (1982) and Juan (1982), the authors showed that the partitions given by the former algorithm are usually better than those provided by the later one for the criterion of average of average (but usually worse for the minimum average) between pairs of clusters dissimilarity.

2.2. Nonhierarchical Algorithms

Differently to the class of hierarchical algorithms, the methods belonging to the heuristic class are designed to cluster entities into a single classification of a known number of clusters which is specified a priori or is determined as part of the clustering method. The central idea in most of these methods is to choose some initial partition of the entities and then alter cluster memberships so as to obtain a better partition. The various algorithms which have been proposed differ as to what constitutes a “better partition” and what methods may be used for achieving improvements. In most cases a technique for stablishing an initial partition is given as part of the original algorithm, but it is usually provided as a convenience to the user than as an integral part of the algorithm. Representative samples of initial partitioning generators are described by Anderberg (1973).

The heuristic methodologies may be used with much larger problems than the hierarchical ones because they do not require to store the data set. The simplest iteractive clustering methods consist of alternating both processes of computing a set of seed points as the centroids of a set of clusters and of constructing a set of clusters by assigning each entity to the cluster with the nearest seed point. Among these methodologies it should be cited the Forgy's Method and the Jancey's Variant, and the MacQueen's k-means Method [1, 5]

The term "k-means" was used by McQueen (1967) to denote the process of assigning each entity to that cluster (among k clusters) with the nearest centroid or mean. By using the first k data units as seed points and relying on only one reallocation pass, this method achieves the distinction of being the least expensive of all heuristic and hierarchical clustering methods cited above. The total effort from the initial configuration trough to the final clusters involves only $k(2n - k)$ distance computations, $(k - 1)(2n - k)$ distance comparisons, and $(n - k)$ centroid updates. This computational workload is only a small fraction of that involved in a hierarchical cluster analysis because k is usually much smaller than n. Apparently the method of McQueen gives useful results because most major changes in cluster memberships occur with the first reallocation. However, a convergent clustering method using the k-means process can be implemented by mainly introducing a continued updating of the clusters until no data units change their cluster memberships. The Forgy, Jancey and convergent k-means methods all use variations on one central process and they exhibit hardly any differences in computational workload. These three methods also converge in the same way, so a final set of clusters produced by one method would satisfy the convergence criteria of the other methods [1].

In one of his works, Milligan (1980) conducted an avaluation of several clustering methods. Artificial clusters were generated in a Euclidian Space in which the approach adopted was to combine the external isolation characteristic with the internal cohesion properties of multivariate normal mixtures. The simulated data sets were clustered by those four nonhierarchical heuristic methodologies cited above and eleven agglomerative hierarchical ones among them the Single Linkage, Complete Linkage, Group Average and Ward's methods. An issue which requires attention is the nature of the starting configuration used for the nonhierarchical methods. First, randomly selected data units were used as seed points; secondly, the centroids obtained from a hierarchical clustering of the data were used as starting centroids. For the purposes of this study the external criterion of Rand (1971) served as the measure of true cluster recovery. All four nonhierarchical heuristic algorithms produced recovery mean values about .90 which were significantly worse than 1.0 of the hierarchical methods in the error-free condition when random starting seeds were used. When rational starting procedures were used, these methods produced recovery values which were equivalent or superior to the hierarchical methods in all error conditions. Thus, the starting partition must be close to the final solution if the heuristic algorithms are to be expected to give good recovery. The Group Average method did place among the top three methods in eight out of ten error condition levels in the experiment. Within the heuristic group, Jancey's method seems to be about the best procedure in most error conditions.

Usually heuristics are applied because even when seemingly small problems are stated, their mathematical programs are large, combinatorial, NP-complete models. As mentioned above, while heuristic approaches are efficient in terms of computational workload, they do not produce optimal clusters if random starting seeds must be used. As local search methods, they proceed by performing a sequence of local changes in an initial solution which improve each time the value of the objective function until a local optimum is achieved. That is, at each iteration an improved solution in the neighborhood of the current solution is obtained until no further improvements are found.

In recent years, several metaheuristics have been proposed which extend in various ways the scheme of nonhierarchical heuristic algorithms and avoid being trapped in local optima with a poor function value [19, 11]. Using systematically the idea of changing of neighborhood in the heuristic search, P. Hansen and N. Mladenovic' proposed a new metaheuristic named Variable Neighborhood Search (VNS for short). Contrary to other metaheuristics based on local search methods, VNS does not follow a trajectory, but explores increasingly distant neighborhoods of the current solution and jumps from there to a new one if an improvement has been made. Among several applications, the VNS was also proposed for solving the Minimum Sum-of-Squares Clustering Problems (MSSC for short) and some results were published by the authors showing the quality of the solutions obtained by VNS to be superior to the quality of the solutions presented by multistart versions of k-Means, H-Means - a nonhierarchic clustering algorithm that is described in section 3, and by a composition of both of them [11]. In this work, a new descent local search heuristic, called J-Means, was proposed; at each step, the cluster centroid is relocated at some entity which does not already coincide with a centroid. This move may be a large one and corresponds to several K-Means moves. Computational study compares six local search methods based on equivalent CPU time; i.e', each heuristic is restarted until a given time elapses. The results show the VNS is the best overall for solving test problems as (i) the 3-dimensional 89-Bavarian postal zones of Spath (1980), (ii) the 4-dimensional 150 Iris of Fisher (1936), (iii) 1060 and (iv) 3038 points in a plane taken from TSP-LIB data base (Reinolt, 1991).

3. Description of the Algorithms

The objective of this section is briefly to describe the new algorithms proposed by Hansen and Mladenovic' [11] and Hansen et al [9] for clustering in order to be possible to note existing differences between them and classical ones. The first subsection describes the VNS Metaheuristic, the J-Means local search and a version of H-Means, both latest for solving the MSSC problems; and the second one describes the divisive hierarchical algorithms developed by Hansen et al.

3.1. Algorithms for MSSC

As mentioned in the earlier section, the main idea in most of heuristic methodologies for clustering consists of alternating both process of computing centroids of a set of clusters and of constructing a set of clusters by assigning each entity to the cluster with the nearest centroid; then, new centroids and objective function values are updated. Let one defines a "neighborhood" of the current solution by all possible such exchange.

1. The VNS metaheuristic comprises the following steps:

   Let us denote a finite set of pre-selected neighborhood structures by N_i , $(i = 1, ..., i_{max}$) and by $N_i(x)$ the set of solutions in the i^{th} neighborhood of x.

   Initialization: Select the set of neighborhood structures N_i ,$i = 1, .., i_{max}$ and find an initial solution x. Choose a stopping condition;

   Repeat the following steps until the stopping condition is met:

   (1) Set $i \longleftarrow 1$;

   (2) Until $i = i_{max}$, repeat the next three steps:

   - Shaking: generate a point x' at random from the i^{th} neighborhood of x ($x' \in N_i(x)$);
   - Local search: apply some local search method with x' as initial solution; denote by x'' the obtained local minimum;
   - Move or not: if this local optimum is better than the incumbent, move there ($x \longleftarrow x''$) and continue the search with N_1 , (i $\leftarrow$ 1); otherwise set $i \longleftarrow i + 1$.

   The stopping condition may be e.g. maximum CPU time allowed, maximum number of iterations, or maximum number of iterations between two improvements. In their work, the authors pointed out that point x' is generated at random in step 2 in order to avoid cicling, which might occur if any deterministic rule is used. It is worth noting that several variants of the VNS metaheuristic can be obtained by using different local search heuristic algorithm in the third part of the step (2). Clearly, the performance of the VNS Metaheuristic depends upon the number of neighborhood it might visit and explore, and, importantly, the "quality" of the defined neighborhood.

2. The heuristic H-Means

   The iterations of the H-Means heuristic consist of alternating entity allocation and centroid relocation; so, it is similar to Cooper's Alternate heuristic for the Multisource Weber Problem (1963). Its main steps can be described in the following way:

   (1) Let X be a set of n entities, $\{C_i$, i=1,...,k$\}$ and $\overline{x}_i$, i=1,...,k, its initial partition and corresponding centroids;

(2) Assign each entity x_j, j=1,...,n, to its closest centroid $\overline{x}_i$, i=1,...,k;

(3) If no change in assignments occurs, a locally optimal partition is found and stop;

(4) Update centroids $\overline{x}_i$ of each cluster C_i and return to step (2)

It should be noted that the solution obtained by H-Means can be improved by heuristic k -Means. In the later one, reassignments of one entity at a time are considered and the centroids of the new clusters can be easily obtained from updating formulas. It should also be noted that the step (1) described above is strongly different to the k-Means first step.

3. The heuristic J-Means

In some problem instances, particularly when the number of clusters is large, existing points, or entities, could be centroids of some clusters in the current solution. These points are refered as "occupied points".

In order to get a neighboring solution of the current one, the centroid $\overline{x}_i$ of a cluster C_i is relocated to some unoccupied entity location and all entities relocated to that cluster centroid. All possible such moves constitute the "jump" neighborhood of the current solution.

It is possible to describe the main steps of the method as follows:

(1) (Initialization)

- Let PM = $\{C_i$, i=1,...,k$\}$, $\overline{x}_i$, i=1,...k, and f_{opt}, be the initial partition of the set X, the corresponding centroids, and the current objetive function value, respectively;

(2) (Occupied points)

- Under appropriate criterion, find unoccupied points, i.e., entities which do not coincide with a cluster centroid;

(3) (Jump Neighborhood)

Find the best partition PM' in the jump neighborhood of the current solution PM in the following way:

- Add a new cluster centroid $\overline{x}_{k+1}$ at some unoccupied entity location x_j and find the index "i'' of the best centroid deletion;
- For each point find its closest centroid and calculate the change in the objective function value (optionally, improve the solution by using some fast local search heuristic);
- Keep the pair (i',j') for which the improvement is better and denote the corresponding solution and objective function value with PM' and f' respectively;

(4) (Termination or move)

- If $f' > f_{opt}$, stop - a local minimum was found in previous iteration (optionally, an improvement of the obtained solution could be possible by using some other heuristic before stoping);
- Otherwise, move to the best neighboring solution PM' (PM = PM'), set $f_{opt} \longleftarrow f'$ and return to step (2).

It should be noted that the main step of J-Means is the step (3) and its relocation part, the first one, can be implemented in O(n). Several variants of this heuristic can be developed, one of them by using k -Means and/or H-Means to improve each jump neighborhood solution or some other heuristic.

The sections 4 and 5 present computational results that are obtained from compositions of VNS metaheuristic with the local search algorithms H-Means and k -Means, named by VNS+H+k -Means, and VNS with the new local search J-Means, the later denoted by VNS+J-Means. In the former composition it is supposed k -Means should be applied in order to improve the solution obtained by H-Means.

3.2. Divisive Hierarchical Algorithms

The two divisive hierarchical algorithms tested in this work have differences on the criterions to be optimized. Both schemes utilize the Tabu Search heuristic for optimizing the local criterion for clustering [9]. The Tabu Search heuristic was adapted to average-linkage divisive hierarchical clustering scheme with the aim of solving an unconstrained quadratic problem in 0-1 variables which defines the bipartition problem for the chosen cluster at each iteration. In order to get an improved initial solution the heuristic may be run several times with different initial random solutions.

The main part of any average-linkage divisive clustering method is a heuristic or exact method for solving the problem of bipartitioning the selected cluster at each iteration. Because this problem is NP-hard, and it is difficult to be solved in practice, instances of large data sets cannot be expected to be solved exactly in reasonable computing time.

The criterions to be optimized by the algorithms are defined as follows:

- D_M - average of between cluster average dissimilarities. This criterion, to be maximized, is an average case type of measure of separation between clusters;
- d_M - minimum interior variance.

These criterions were chosen by the authors because, contrary to other global criteria used in clustering, they do not suffer from distorsion effects due to the size of the clusters [9] such as the within-clusters sum of dissimilarities.

The algorithms that utilize the D_M and d_M criterions will be called AVHT and VAHT, respectively.

4. Computational Results

The algorithms described in earlier section have been running on the SPARC SUN STATION IPX at GERAD - Groupe d'Etudes et de Recherche en Analyse des Decisions, Montreal, Canada, with exeption to the hierarchical agglomerative one. The later algorithm is a version of the Group Average methodology and was obtained from a package by G. Milligan and R. Cheng, Ohio State University, 1993, which requires DOS environment to be run. The parameters that are used to measure and compare the performances of these algorithms are of Hubert and Arabie and, separately, of Rand external recovery measurements, what can explains the possibility of running the studied algorithms in different computational enviroments.

Six algorithms are analysed in this work: three metaheuristics, two hierarchical divisive and one hierarchical agglomerative ones which are listed below.

1. multistart k-Means;
2. metaheuristic VNS composed with H-Means and k -Means, which is named VNS+H+k-Means;
3. metaheuristic VNS using J-Means as a local search, which is named VNS+J-Means;
4. average-linkage hierarchical divisive AVHT described in section 3.2;
5. average-linkage hierarchical divisive VAHT described in section 3.2;
6. hierarchical agglomerative Group Average.

The softwares of the three first algorithms were obtained from their authors [11], the forth and fifth algorithms were run from SPHINX1 package at GERAD, and the last one is a version by G. Milligan and revised by R. Cheng.

Before displaying the data results on tables, it is important to make known the procedure of initialization that was applied for each of those algorithms. Metaheuristics 1, 2 and 3, as mentioned above, are initialised from a solution obtained by random. Parameters defining an upper bound for the number of generated initial solutions are necessary to run the multistart k-Means algorithm as well as to obtain the number of neighborhood to be explored by the metaheuristic VNS, and also to control the process of changing neighborhood. For the multistart k-Means it was used a parameter that controlled the time for generating and comparing initial solutions among the best one should be chosen. For the tests displayed on tables below the time parameter was fixed to be 0.5 seconds for n = 50 (Table 1), and 2.0 seconds for n = 200 (Table 2).

It is worth pointing out that the version VNS+J-Means algorithm uses in a first and unique iteration the heuristic k-Means to obtain an initial solution for the local search procedure J-Means. Relating to the hierarchical divisive algorithms refered in section 3.2, the initialization process of the adapted heuristic Tabu Search was also a random one. The hierarchical agglomerative algorithm Group Average of Lance and Williams is a classical one and, as well-known, it is initialized by merging a new cluster from the two “closest” clusters among all existing ones in a fixed iteration, having the measure of similarity characterized by the average of all links within the merged cluster [1].

For the results that are displayed in Table 1 and Table 2 below, samples of distinct sizes were generated by a generator program by G.W. Milligan [17] exhibiting the properties of internal cohesion and external isolation. The data generation process used in the present study corresponds to the error-free data conditons, where the distribution of points within clusters follows a truncated multivariate normal. In this way, it was obtained a data set of 72 samples with 50 entities and another one of 72 samples with 200 entities, which were embedded in either a m=4 or m=6 dimensional Euclidian space. The generated samples with 50 entities were distributed into groups presenting true cluster structure containing either 2, 3 or 4 clusters. Twelve different samples were generated and tested for each group characterized by both the true number of clusters and the characteristic number of entities. A similar procedure was applied to generate data samples with 200 entities differing from the former in terms of both the true number of clusters and the number of generated data samples per each group. In the last case, samples were generated presenting cluster structure containing either 2, 3, 4, or 5 clusters, and nine different data samples were generated and tested for each group characterized by both the true number of clusters and the characteristic number of entities. Thus, each line of Table 1 and Table 2 below shows the mean recovery value of Hubert and Arabie external measurement that had been found for those data samples with 50 and 200 entities, respectively.

$m = 4^*$						
	Metaheuristics			**Divisive**		**Agglom.**
k	**k-Mean**	**VNAS+H+ k-Means**	**VNS+J-Mean**	**VAHT**	**AVHT**	**Group Average**
2	.937	.937	.937	1.0	.677	1.
3	.91	.91	.91	.586	.607	.77
4	.85	.85	.85	.554	.628	.77
mean	.90	.90	.90	.713	.637	.846
$m = 6^*$						
2	.99	.99	.99	.95	.807	1.
3	1.0	1.0	1.0	.668	.826	.97
4	.75	.75	.746	.44	.666	.68
mean	.913	.913	.912	.686	.766	.883
total	.90	.90	.90	.699	.701	.864

Table 1: Mean recovery values for data samples with 50 entities

By observing the results displayed on Table 1, it is clear that the recovery value presented by all of the studied algorithms worsens when increasing the number of clusters, and generally improves when a larger number of characteristics is used in exception to VAHT algorithm. This results are compatible to the analysis presented by Milligan and Sokol (1983). Concerning to the number of characteristics, it is not possible to have the same conclusions when observing the recovery values displayed on Table 2 below. For all algorithms, excepting AVHT, the mean values obtained for m =4 presented better recovery than those ones obtained for $m = 6$.

Clearly, VAHT and AVHT divisive hierarchical algorithms presented worst performance when compared to those metaheuristics ones. For two clusters VAHT algorithm performance is very good, and its recovery is perfect for $m = 4$. Metaheuristic algorithms showed to have, among themselves, similar performances for any number of characteristics and any number of clusters. Moreover, it is worth pointing out the good performances obtained by them for data samples with true structure of either two or three clusters, in any case. An explanation for this similarity can be found by observing their initialization process and the importance of k -Means methodology for all metaheuristics studied in this work. However, it is possible to observe little difference between their performances for larger data samples as it is displayed on Table 2.

$m = 4^*$						
	Metaheuristics			**Divisive**		**Agglom.**
k	**k-Means**	**VNAS+H+ k-Mean**	**VNS+J- Mean**	**VAHT**	**AVHT**	**Group Average**
2	1.0	.915	.915	.905	.640	1.
3	.99	.99	.99	.602	.678	.97
4	.826	.840	.826	.577	.582	.79
5	.89	.91	.91	.462	.620	.86
mean	.926	.913	.910	.636	.630	.905
$m = 6^*$						
2	.914	.914	.996	.996	.671	.99
3	.954	.954	.954	.671	.865	.99
4	.820	.785	.825	.473	.576	.85
5	.673	.646	.648	.390	.486	.81
mean	.840	.824	.855	.633	.649	.885
total	.883	.868	.882	.634	.639	.895

Table 2: Mean recovery values for data samples with 200 entities

In many cases metaheuristic methodologies presented better mean recovery values than the hierarchical agglomerative one, but more detailed observations should be done:

- the divisive VAHT and the agglomerative Groupe Average showed to have better recovery index than the metaheuristics for samples with only two clusters independent on the sample size and the characteristic number of entities;

- Group Average algorithm presented the best recovery index for $n = 200$, $m = 6$ and any k;
- metaheuristic algorithms exibited better recovery index than the hierarchical ones for cluster numbers higher than two;
- the hierarchichal divisive VAHT and AVHT algorithms showed to be the worst for increasing number of clusters.

The results displayed on both tables suggest the superiority of multistart k-Means and VNS metaheuristics over the other metodologies when increasing number of clusters are considered. These results can be explained by noting the good capability of VNS of exploring farthest neighborhoods of a current solution, and the efficiency of k-Means to converge when starting solutions are closed to the global one. Furthermore, the heuristics k-Means, H-Means and J-Means have good flexibility of changing cluster structures at each step.

Despite its simple multistart random initialization, which does not control explorations of farthest neighborhoods as VNS does, the k-Means algorihm appears aside to the other metaheuristics. The simplicity of initializing multistart k-Means turns the algorithm the most interesting among the six tested in this work when considering both together implementation complexity and attainment of good results. The superiority of the Group Average over the other algorithms for $n = 200$ and $m = 6$ can found explanation by noting that it invokes the mean instead of the variance criterion, and this fact appears to influence the algorithm to become more suitable when the clusters are of unequal size [18].

Tables 1 and 2 show hierarchical divisive VAHT and AVHT algorithms are clearly worse than metaheuristics and hierarchichal agglomerative ones. Just VAHT algorithm presented good performance for data samples with true structures of two clusters. These not good results presented by VAHT and AVHT can be explained by attesting both facts: they were implemented into a non-exact TABU heuristic of searching new clusters at each stage, and, furthermore, they have the property of acumulating errors into their stages.

Some comments concerned to the Rand measurement of recovery are also made in the next section, with the aim of comparing the results obtained by Milligan [15] and the results obtained by the metaheuristics studied in this work.

5. Conclusion and Discussion

The recovery index of Rand was also computed for all of the six algorithms studied in this work. As expected the analysis of algorithm performances does not change when applying the Rand recovery measurement. The total average of indexes are obtained: .95 for multistart k-Means, .944 for VNS+H+k-Means, .951 for VNS+J-Means and .954 for Group Average algorithms. By using the same sample generator program with the "error free condition", G.W. Milligan [15] showed in his work that Group Average algorithm presented the best Rand recovery index among thirteen different hierarchical agglomerative and four different nonhierarchical clustering algorithms. In this work, a single starting k-Means heuristic algorithm was listed among those nonhierarchical ones, and its performance placed among the worst when starting solutions were not enough closed to the global one. So, it is worth underlining the possibility of geting very different results from k-Means algorithm when not considering a multistart process.

In the present work the metaheuristics showed to get good results even for samples not large and few number of clusters. Interestingly, the version of multistart k-Means presented performance similar to the new VNS. It seems this similarity could be explained by noting both the importance of local k-Means methodology for the structure of all metaheuristics studied and their initialization processes. Undoubt, for the data samples considered, simple random multistart procedure has generated good starting solutions for k-Means algorithm.

As few researchers already pointed out, algorithms for clustering should have their performances more frequently analysed from computing recovery index values. The reasons of using external measurements without underlying the computational workload of the algorithms have statistical support interests: in this case, more accurate solution can means to get a much better cluster structure. In cluster analysis computational workload does not matter when considering both good classical algorithm and problems with small-medium size. Metaheuristics have benn proposed with the aim of improving resolutions of large problems where classical algorithms do not work well. Fortunately, the results presented in this work showed metaheuristics that are implemented into the classical heuristic k-Means have very good room, even for problems that had already been solved efficiently by the hierarchical agglomerative Group Average. Adding to the obtained results the fact that the local heuristics k-Means, H-Means and J-Means have good flexibility of changing cluster structure at each step, it is expected VNS metaheuristics could also reach good solutions for problems presenting different cluster structures of those generated in this work. Particularly, it should be a interesting work to develop a recovering comparative study between the multistart k-Means, VNS metaheuristics and hierarchical agglomerative clustering algorithms by exploring different structures of clusters.

References

[1] M.R. Anderberg, "Cluster Analysis for Applications", N.Y., Academic Press, 1973.

[2] J.P. Benzecri, "Construction d'une Classification Ascendante Hierarchique par la Recherche en Chaine des Voisins Reciproques", *Les Cahiers de l'Analyse des Donnees*, **VII** (2) (1982), 209-218.

[3] G. Diehr, Evaluation of a Branch-and-Bound Algorithm for Clustering, *SIAM J. SCI. Stat. Comp.,* **6** (2) (1985).

[4] B. Everitt, "Cluster Analysis", (Third Edn.) London, Heinemann Educational Books, 1993.

[5] A.D. Gordon, "Classification: Methods for the Exploratory Analysis of Multivariate Data", London, Chapman and Hall, 1981.

[6] A.D. Gordon, A Review of Hierarchical Classification, *J. R. Statist. Soc. A.,* **150** (2) (1987), 119-137.

[7] A. Guenoche, P. Hansen and B. Jaumard, Efficient Algorithms for Divisive Clustering with the Diameter Criterion, *J. of Classification*, **8** (1991), 5-30.

[8] D.J. Hand, "Discrimination and Classification", John Wiley and Sons, Chichester, 1981.

[9] P. Hansen, B. Jaumard and E. da Silva, "Average-Linkage Divisive Hierarchical Clustering", Les Cahiers du GERAD, G-91-55, Montreal, Canada, 1993.

[10] P. Hansen and B. Jaumard, Cluster Analysis and Mathematical Programming, *Math. Programming,* **79** (1997), 191-215.

[11] P. Hansen and N. Mladenovic', "J-Means: A New Local Search Heuristic for Minimum Sum-of-Squares Clustering", Les Cahiers du GERAD, G-99-14, feb.1999.

[12] J.A. Hartigan, "Clustering Algorithms", New York, Wiley, 1975.

[13] J. Juan, Programme de Classification Hierarchique par l'Algorithme de la recherche en Chaine des Voisins Reciproques, *Les Cahiers de l'Analyse des Donnees*, **VII** (2) (1982), 219-225.

[14] G. Klein and J.E. Aronson, Optimal Clustering: A Model and Method, *Naval Research Logistics*, **38** (1991), 447-461.

[15] G.W. Milligan, An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms, *Psychometrika*, **45** (3) (1980).

[16] G.W. Milligan, A Study of the Comparability of External Criteria for Hierarchical Cluster Analysis, *Multivariate Behavioral Research*, **21** (1986), 441-458.

[17] G.W. Milligan, An Algorithm for Generating Artificial Test Clusters, *Psychometrika*, **50** (1) (1985), 123-127.

[18] G.W. Milligan and L. M. Sokol, The Effect of Cluster Size, Dimensionality, and the Number of Clusters on Recovery of True Cluster Structure, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-5** (1) (1983).

[19] N. Mladenovic' and P. Hansen, Variable Neighborhood Search, *Computers Ops. Res.*, **24** (11) (1997), 1097-1100.

[20] F. Murtagh, A Survey of Recent Advances in Hierarchical Clustering Algorithms, *The Computer Journal*, **26** (4) (1983).

[21] M. R. Rao, Cluster Analysis and Mathematical Programming, *J. of the American Statistical Association*, **66** (1971), 622-626.