

# COMBINATORIAL OPTIMIZATION HEURISTICS IN PARTITIONING WITH NON EUCLIDEAN DISTANCES

Eduardo Piza<sup>1</sup>, Javier Trejos<sup>2</sup>, y Alex Murillo<sup>3</sup>, CIMPA, Escuela de Matemática, Universidad de Costa Rica, San José, Costa Rica

## ABSTRACT

We study some criteria that can be applied for the partitioning of a set of objects when non Euclidean distances are used; particularly, these criteria can be used when the data are described by binary variables. These criteria are based on aggregations that measure the homogeneity of a class and some are generalizations of variance or inertia. Properties of the criteria are studied and partitioning methods are proposed, based on metaheuristics of global optimization, such as simulated annealing and tabu search. Finally, comparative results on binary data are shown.

**Key words:** binary data, qualitative data, clustering, automatic classification, simulated annealing, tabu search, generalized inertia.

## RESUMEN

Se estudian criterios que se pueden aplicar para particionar un conjunto de objetos cuando se usan distancias no euclídeas; en particular, los criterios pueden ser usados cuando los datos son descritos por variables binarias. Estos criterios están basados en agregaciones que miden la homogeneidad de una clase y algunos son generalizaciones de la varianza o inercia. Se estudian algunas de las propiedades de los criterios de agregación y se proponen métodos de particionamiento basados en el uso de metaheurísticas de optimización global, como sobrecaleamiento simulado y búsqueda tabú. Finalmente, se muestran resultados comparativos sobre datos binarios.

**Palabras clave:** datos binarios, datos cualitativos, análisis de conglomerados, clasificación automática, búsqueda tabú, sobrecaleamiento simulado, inercia generalizada.

MSC: 90C27

## 1. INTRODUCTION

Usual methods of partitioning (Forgy, k-means, dynamical clusters transfers, Isodata, etc.) find local optima of the inertia (or general variance) criterion since they are based on procedures of local search (Anderberg (1973), Bock (1974), Diday et.al (1982)). In the Euclidean case, the authors have employed combinatorial optimization techniques, such as simulated annealing, tabu search and evolutionary strategies for optimizing the criterion, obtaining excellent results (see Trejos *et al.* (1998) or Piza *et al.* (1999)). In the case of non quantitative data, or if one wants to use a non Euclidean distance, it is necessary to adapt the criterion, since Huygens theorem and other theoretical results hold only in an Euclidean context. The use of the  $L_1$  distance in the quantitative case has been considered in Jajuga (1987) or Späth (1985) and it is proved that the best center adapted to a class is the vector of variable medians.

In this paper we study six aggregation indexes that can be used for measuring the homogeneity or compactness of a partition when general dissimilarity indexes are used. We deal with some theoretical properties of these aggregations, such as monotonicity and up-downdating formulas when methods of transfers are used. Also, a Huygens-like property is deduced.

## 2. THE PROBLEM OF PARTITIONING IN A NON-EUCLIDEAN CONTEXT

Let  $D = (d(x, x^1))_{n \times n}$  be a dissimilarity matrix on a set of  $n$  objects  $\Omega = \{x_1, x_2, \dots, x_n\}$ .

We seek for a partition  $P = (C_1, C_2, \dots, C_k)$  of  $k$  classes of objects, and a numerical criterion  $W$  must be defined for measuring the quality of the partition. That is, we want to solve the problem

<sup>1</sup>E-mail: [epiza@cariari.ucr.ac.cr](mailto:epiza@cariari.ucr.ac.cr) Fax: +(506) 207 4397

<sup>2</sup>E-mail: [jtrejos@cariari.ucr.ac.cr](mailto:jtrejos@cariari.ucr.ac.cr) Fax: +(506) 207 4397

<sup>3</sup>E-mail: [murillo@cariari.ucr.ac.cr](mailto:murillo@cariari.ucr.ac.cr) Fax: +(506) 556 7020

$$\min_{P \in P_k} W(P) = \sum_{j=1}^k \delta(C_j) \quad (1)$$

where  $P_k$  is the set of all partitions of  $\Omega$  in  $k$  or less classes, and  $\delta(C)$  is some aggregation measure. Objects in  $\Omega$  may be described by a set of variables, not necessarily numerical (binary or categorical). In this case, there are several dissimilarity indices that can be used, such a Jaccard, Dice or Russel-Rao in the binary case, and  $X^2$  or Hamming in the categorical case.

Usually one uses the within-inertia criterion for measuring the homogeneity of the partition (see Bock (1974) and Diday **et al.** (1982)). Classes are represented by a "center" which is the centroid of the class, that is the mean vector in the Euclidean space or the median vector in the  $L_1$  space. In the context of non-quantitative data, Diday (1980) has proposed to use a "center" that minimizes the sum of distances to the rest of elements of the class, and to proceed with the dynamical clusters algorithm. We do not follow this way since the dynamical clusters algorithm finds local minima and also because a center in this context can be hard to find and may be an element without representative sense, as often occurs in practice.

For the clustering of  $\Omega$  there are three ways to proceed:

- To use the classical *hierarchical classification* theory (see Bock (1974), Diday **et al.** (1982) or Piza (1987)) with a dissimilarity index and one of the aggregations adapted in the non Euclidean case (for example, single linkage, complete linkage or average linkage in the agglomerative approach).
- To use the multidimensional scaling (MDS) theory (see for example Borg and Groenen (1997), Trejos and Villalobos (in press) or Villalobos and Trejos (in press) in this volume). Metric MDS finds a configuration in an Euclidean space so that Euclidean distances are as near as possible as the original dissimilarities. Then, classical clustering methods can be used for finding a partition of  $\Omega$ .
- To define a good aggregation index  $\delta$  and solve the problem using an adapted heuristic for partitioning.

We proceed in the last way, studying some aggregation indexes  $\delta$  and proposing algorithms based on well known techniques of optimization.

### 3. AGGREGATION INDEXES

A dissimilarity index  $d$  on  $\Omega$  satisfies:

- (i) Symmetry:  $d(x, x') = d(x', x)$ ,  $\forall x, x' \in \Omega$ .
- (ii)  $d(x, x) = 0$ ,  $\forall x \in \Omega$ .

We will suppose that  $d$  is a dissimilarity index defined on  $\Omega$ . We study the following aggregation indexes defined on  $2^\Omega$ : for a class  $C \subseteq \Omega$ , let  $|C|$  be the number of objects in  $C$  and define:

a. Single linkage:

$$\delta_1(C) = \min\{d(x, x') : x, x' \in C\}$$

b. Complete linkage: \*

$$\delta_2(C) = \max\{d(x, x') : x, x' \in C\}$$

c. Sum of the dissimilarities:

$$\delta_3(C) = \sum_{\substack{x, x' \in C \\ x \neq x'}} d(x, x')$$

d. Späth aggregation:

$$\delta_4(C) = \frac{1}{2|C|} \sum_{\substack{x, x' \in C \\ x \neq x'}} d(x, x')$$

e. Average of the dissimilarities:

$$\delta_5(C) = \frac{1}{|C|(|C|-1)} \sum_{\substack{x, x' \in C \\ x \neq x'}} d(x, x')$$

and  $\delta_5(C) = 0$  if  $|C| = 1$

f. Variance of the dissimilarities:

$$\delta_6(C) = \frac{1}{|C|(|C|-1)} \sum_{\substack{x, x' \in C \\ x \neq x'}} [d(x, x') - \mu(C)]^2,$$

where  $\mu(C) = \delta_5(C)$  is the average of the dissimilarities, we define  $\delta_6(C) = 0$  if  $|C| = 1$ .

### 3.1. Properties

The following property is necessary for consistency of the partitions obtained in automatic classification.

**Definition 1. (Monotonicity property)** Let  $P = (C_1, \dots, C_k) \in P_k^*$  and  $P' = (C'_1, \dots, C'_{k+1}) \in P_{k+1}^*$  be partition of  $\Omega$  in  $k$  and  $k+1$  non-empty classes, respectively. We say that the aggregation index  $\delta$  on  $2^\Omega$  satisfies the monotonicity property, if for all instances of the data, we have

$$\min_{P' \in P_{k+1}^*} W(P') = \sum_{j=1}^{k+1} \delta(C'_j) \leq \min_{P \in P_k^*} W(P) = \sum_{j=1}^k \delta(C_j),$$

for every number of classes  $k < n$ . That is, the value of the objective function  $W(P)$  of the solution of the optimization problem for  $k+1$  classes is no greater than the corresponding value of the optimization problem for  $k$  classes.

**Remark :** Here  $P_k^*$  denotes the set of all partitions of  $\Omega$  in  $k$  non-empty classes, while  $P_k$  is the set of all partitions of  $\Omega$  in  $k$  or less classes.

**Theorem 1.** All aggregations  $\delta_1, \dots, \delta_6$  satisfy the monotonicity property.

**Proof:** Let us consider the partition  $\hat{P} = (C_1, \dots, C_k)$  is  $k$  non-empty classes, solution of the optimization problem of  $\min \{W(P) : P \in P_k\}$ . Then, it is enough to construct, from  $\hat{P}$ , a new partition  $Q$  in  $k+1$  non-empty classes, such that  $W(Q) \leq W(\hat{P})$ : beginning with  $\hat{P}$  we can construct  $Q$  by transferring one of the objects  $x_i \in \Omega$  from a non-unitary class of  $\hat{P}$  (say the first class  $C_1$  to a new unitary class or singleton:  $Q = (C_1 \setminus \{x_i\}, C_2, \dots, C_k, \{x_i\})$ . The reader can easily precise the way to choose  $x_i \in \Omega$  (in each case it depends on the aggregation index  $\delta_1, \dots, \delta_6$ ) such that  $W(Q) \leq W(\hat{P})$ .  $\square$

**Theorem 2 (Single linkage partition).** Let  $x_p$  and  $x_q$  be objects of  $\Omega$  with minimal dissimilarity. Then, a solution to the optimization problem corresponding to the index  $\delta_1$  (single linkage),  $\min_{P \in P_k} W(P) = \sum_{j=1}^k \delta_1(C_j)$ , is

obtained for the partition  $P^* = (C_1^*, \dots, C_k^*)$  of  $\Omega$  in  $k$  classes, which has the first  $k-1$  unitary classes and the last class  $C_k^*$  of size  $n-k+1$ , where  $\{x_p, x_q\} \subseteq C_k^*$ .

**Proof:** For any partition  $P = (C_1, \dots, C_k)$  of  $\Omega$  in  $k$  classes, we have

$$W(P) = \sum_{j=1}^k \delta_1(C_j) = \sum_{j=1}^k \min \{d(x, x') : x, x' \in C_j\} \leq \min \{d(x, x') : x, x' \in \Omega\} = d(x_p, x_q).$$

Then, it is enough to choose any  $k - 1$  objects of  $\Omega$ , different from  $x_p$  and  $x_q$  and to distribute them in the initial  $k - 1$  classes  $C_j$ , constructing singletons. The partition  $P^* = (C_1^*, \dots, C_k^*)$  constructed by this way is such that  $W(P^*) = d(x_p, x_q)$ . ■

**Theorem 3** Let  $P_{opt}$  be the set of all solutions to the optimization problem.

$$\min_{P \in P_k} W(P) = \sum_{j=1}^k \delta(C_j).$$

Then, for all aggregation indexes  $\delta_1, \dots, \delta_6$ , there exist a solution  $P^* \in P_{opt}$  that has non empty classes. Moreover, the aggregation indexes  $\delta_3$  (sum of dissimilarities) and  $\delta_4$  (Späth aggregation), all partitions  $P \in P_{opt}$  have non empty classes.

**Proof:** The theorem holds for aggregation  $\delta_1$ , as it was already proved. For the other aggregations  $\delta_r$ , con  $r \in \{2, \dots, 6\}$ , let  $P = (C_1, \dots, C_k) \in P_{opt}$  and suppose that  $P$  has an empty class. Let us see how to "fill" that class by transferring an object  $x_i$ , chosen in any of the other non unitary classes, say from class  $C$ , in such a way that inequality  $\delta_r(C \setminus \{x_i\}) \leq \delta_r(C)$  is satisfied. If the inequality is strict, then we obtain a contradiction to the fact that  $P \in P_{opt}$ , and thus  $P_{opt}$  does not have partitions with empty classes. We use the recursive formulae shown later in theorem 5 for the computation of  $\delta_r(C \setminus \{x_i\})$ .

(a) Aggregation  $\delta_2$ : it can be chosen any  $x_i \in C$ , since it is always satisfied  $\max\{d(x, x') : x, x' \in C \setminus \{x_i\}\} \leq \max\{d(x, x') : x, x' \in C\}$ , for any  $x_i \in C$ .

(b) Aggregation  $\delta_3$ : any choice of  $x_i \in C$  is useful, since  $\delta_3(C \setminus \{x_i\}) = \delta_3(C) - \sum_{x \in C} d(x, x_i) < \delta_3(C)$ . Remark that the inequality is strict, hence the partition obtained by the transfer of  $x_i$  always improves the criterion.

(c) Aggregation  $\delta_4$ : solving the inequality  $\delta_4(C \setminus \{x\}) \leq \delta_4(C)$ , we obtain the equivalent inequality  $\delta_4(C \setminus \{x_i\}) \leq \sum_{x \in C} d(x, x_i)$  which is always strictly satisfied when we choose the object  $x_i \in C$  that maximizes  $\sum_{x \in C} d(x, x_i)$ .

(d) Aggregation  $\delta_5$ : solving the inequality  $\delta_5(C \setminus \{x_i\}) \leq \delta_5(C)$ , we obtain the equivalent inequality  $\delta_5(C) \leq \frac{1}{|C| - 1} \sum_{x \in C} d(x, x_i)$ . The same choice of any object  $x_i \in C$  that maximizes  $\sum_{x \in C} d(x, x_i)$  is useful. However it must be remarked here that the equality holds when all the dissimilarities between objects of  $C$  are equal.

(e) Aggregation  $\delta_6$ : the inequality  $\delta_6(C \setminus \{x_i\}) \leq \delta_6(C)$  is satisfied at least when we choose any object  $x_i \in C$  that maximizes the quantity  $\frac{1}{|C|(|C| - 1)} \sum_{x \neq x' \in C} [d(x, x') - \mu(C)]^2$ . Also in this case the quantities  $\delta_6(C \setminus \{x_i\})$  and  $\delta_6(C)$  are equal when all the dissimilarities between objects of  $C$  are equal.

This reasoning is repeated by transferring objects from non unitary classes to fill the empty classes of  $P$ , until in the partition  $P^*$  made by this way there are no more empty classes. It is clear that  $W(P^*) \leq W(P)$  and the inequality is always strict when aggregations  $\delta_3$  and  $\delta_4$  are used. It can be also remarked that aggregations  $\delta_5$  and  $\delta_6$  have a tendency to produce optimal partitions without empty classes, with the pointed out exceptions.

**Lemma 4 (Huygens decomposition using  $\delta_6$ )** For any class  $C \subseteq \Omega$  and any real number  $\beta$ , it holds the decomposition:

$$\frac{1}{|C|(|C|-1)} \sum_{\substack{x, x' \in C \\ x \neq x'}} [d(x, x') - \beta]^2 = \delta_6(C) + [\mu(C) - \beta]^2.$$

**Proof:** We have that

$$\begin{aligned} \sum_{\substack{x, x' \in C \\ x \neq x'}} [d(x, x') - \beta]^2 &= \sum_{\substack{x, x' \in C \\ x \neq x'}} [d(x, x') - \mu(C)]^2 + \sum_{\substack{x, x' \in C \\ x \neq x'}} [\mu(C) - \beta]^2 \\ &\quad + 2 \sum_{\substack{x, x' \in C \\ x \neq x'}} [d(x, x') - \mu(C)][\mu(C) - \beta] \\ &= |C|(|C| - 1) \{ \delta_6(C) + [\mu(C) - \beta]^2 \} \\ &\quad + 2[\mu(C) - \beta] \sum_{\substack{x, x' \in C \\ x \neq x'}} [d(x, x') - \mu(C)]. \end{aligned}$$

The last term is clearly null, since  $\mu(C)$  is the mean of all dissimilarities  $d(x, x')$  between objects of  $C$ . ■

### 3.2 Transfers of objects: up and downdating formulae

Many classification methods are based on the transfers of objects from one class to another class. This is the case of the k-means (Anderberg (1973), Bock (1974)) and three methods based on metaheuristics, proposed by the authors in Trejos **et al.** (1998). In the non-Euclidean context, we will also propose methods based on transfers of objects, hence we need to study the up and downdating formulas for the proposed aggregation indexes.

Let  $P = (C_1, \dots, C_k)$  and  $\tilde{P} = (\tilde{C}_1, \dots, \tilde{C}_k)$  be two partitions on  $\Omega$ , such that object  $i$  is transferred from class  $C_j$  to class  $\tilde{C}_\ell$  (this transfer will be noted  $(C_j \xrightarrow{i} \tilde{C}_\ell)$ ). So, we need to calculate the following new aggregation, in terms of the actual values for  $\delta(C_\ell)$  and  $\delta(C_j)$ .

- $\delta(C_\ell \cup \{x_i\})$ : the aggregation of the augmented class, in terms of  $\delta(C_\ell)$ .
- $\delta(C_j \setminus \{x_i\})$ : the aggregation of the reduced class, in terms of  $\delta(C_j)$ .

The following are the up and downdating formulae that we found, for the aggregation indexes  $\delta_1, \dots, \delta_6$ .

**Theorem 5. Recursive formulae for the computation of  $\delta(C_\ell \cup \{x_i\})$ :**

$$\begin{aligned} \delta_1(C_\ell \cup \{x_i\}) &= \min(\delta_1(C_\ell), \min\{d(x, x_i) : x \in C_\ell\}). \\ \delta_2(C_\ell \cup \{x_i\}) &= \max(\delta_2(C_\ell), \max\{d(x, x_i) : x \in C_\ell\}). \\ \delta_3(C_\ell \cup \{x_i\}) &= \delta_3(C_\ell) + \sum_{x \in C_\ell} d(x, x_i). \\ \delta_4(C_\ell \cup \{x_i\}) &= \frac{n_\ell}{n_\ell + 1} \delta_4(C_\ell) + \frac{1}{n_\ell + 1} \sum_{x \in C_\ell} d(x, x_i). \\ \delta_5(C_\ell \cup \{x_i\}) &= \frac{n_\ell - 1}{n_\ell + 1} \delta_5(C_\ell) + \frac{2}{n_\ell(n_\ell + 1)} \sum_{x \in C_\ell} d(x, x_i). \end{aligned}$$

$$\delta_6(C_\ell \cup \{x_i\}) = \frac{n_\ell - 1}{n_\ell + 1} \delta_6(C_\ell) + \frac{n_\ell - 1}{n_\ell + 1} [\mu(C_\ell \cup \{x_i\}) - \mu(C_\ell)]^2 + \frac{2}{n_\ell(n_\ell + 1)} \sum_{x \in C_\ell} [d(x, x_i) - \mu(C_\ell \cup \{x_i\})]^2$$

Here  $n_\ell = |C_\ell|$ . In the last formula we have that  $\delta_6(C_\ell \cup \{x_i\}) = 0$ , when  $n_\ell = 0$ .

**Proof:** Formulae for aggregation  $\delta_1, \delta_2$  and  $\delta_3$  are elementary. We will deal only with the rest:

**Aggregation  $\delta_4$ :**

$$\begin{aligned} \delta_4(C_\ell \cup \{x_i\}) &= \frac{1}{2(n_\ell + 1)} \sum_{\substack{x, x' \in C_\ell \cup \{x_i\} \\ x \neq x'}} d(x, x') \\ &= \frac{1}{n_\ell + 1} \left\{ \sum_{\substack{x, x' \in C_\ell \\ x \neq x'}} d(x, x') + \sum_{x \in C_\ell} d(x, x_i) + \sum_{x \in C_\ell} d(x_i, x) \right\} \\ &= \frac{1}{2(n_\ell + 1)} \left\{ 2n_\ell \delta_4(C_\ell) + 2 \sum_{x \in C_\ell} d(x, x_i) \right\} \\ &= \frac{n_\ell}{n_\ell + 1} \delta_4(C_\ell) + \frac{1}{n_\ell + 1} \sum_{x \in C_\ell} d(x, x_i). \end{aligned}$$

**Aggregation  $\delta_5$ :**

$$\begin{aligned} \delta_5(C_\ell \cup \{x_i\}) &= \frac{1}{(n_\ell + 1)n_\ell} \sum_{\substack{x, x' \in C_\ell \cup \{x_i\} \\ x \neq x'}} d(x, x') \\ &= \frac{1}{(n_\ell + 1)n_\ell} \left\{ \sum_{\substack{x, x' \in C_\ell \\ x \neq x'}} d(x, x') + 2 \sum_{x \in C_\ell} d(x, x_i) \right\} \\ &= \frac{1}{(n_\ell + 1)n_\ell} \left\{ n_\ell(n_\ell - 1) \delta_5(C_\ell) + 2 \sum_{x \in C_\ell} d(x, x_i) \right\} \\ &= \frac{n_\ell - 1}{n_\ell + 1} \delta_5(C_\ell) + \frac{2}{(n_\ell + 1)n_\ell} \sum_{x \in C_\ell} d(x, x_i). \end{aligned}$$

**Aggregation  $\delta_6$ :** we use Huygens lemma 4:

$$\begin{aligned} \delta_6(C_\ell \cup \{x_i\}) &= \frac{1}{(n_\ell + 1)n_\ell} \sum_{\substack{x, x' \in C_\ell \cup \{x_i\} \\ x \neq x'}} [d(x, x') - \mu(C_\ell \cup \{x_i\})]^2 \\ &= \frac{1}{(n_\ell + 1)n_\ell} \left\{ \sum_{\substack{x, x' \in C_\ell \\ x \neq x'}} [d(x, x') - \mu(C_\ell \cup \{x_i\})]^2 + 2 \sum_{x \in C_\ell} [d(x, x_i) - \mu(C_\ell \cup \{x_i\})]^2 \right\} \\ &= \frac{n_\ell - 1}{n_\ell + 1} \delta_6(C_\ell) + \frac{n_\ell - 1}{n_\ell + 1} [\mu(C_\ell) - \mu(C_\ell \cup \{x_i\})]^2 + \frac{2}{n_\ell(n_\ell + 1)} \sum_{x \in C_\ell} [d(x, x_i) - \mu(C_\ell \cup \{x_i\})]^2 \end{aligned}$$

■

**Theorem 6.** Recursive formulae for the computation of  $\delta(C_j \setminus \{x_i\})$ :

$$\delta_3(C_j \setminus \{x_i\}) = \delta_3(C_j) - \sum_{x \in C_j} d(x, x_i).$$

$$\delta_4(C_j \setminus \{x_i\}) = \frac{n_j}{n_j - 1} \delta_4(C_j) - \frac{1}{(n_j - 1)} \sum_{x \in C_j} d(x, x_i).$$

$$\delta_5(C_j \setminus \{x_i\}) = \frac{n_j}{n_j - 2} \delta_5(C_j) - \frac{2}{(n_j - 1)(n_j - 2)} \sum_{x \in C_j} d(x, x_i).$$

$$\delta_6(C_j \setminus \{x_i\}) = \frac{n_j}{n_j - 2} \delta_6(C_j) - [\mu(C_j \setminus \{x_i\}) - \mu(C_j)]^2 - \frac{2}{(n_j - 1)(n_j - 2)} \sum_{x \in C_j \setminus \{x_i\}} [d(x, x_i) - \mu(C_j)]^2.$$

Here  $n_j = |C_j|$ . The value of  $\delta_r(C_j \setminus \{x_i\})$  is 0 when there is division by 0 in the preceding formulae. The dissimilarities  $\delta_1$  and  $\delta_2$  do not have a recursive formula for the computation of  $\delta(C_j \setminus \{x_i\})$ .

**Proof:** The formula for  $\delta_3$  is elementary. We will deal only with the rest.

**Aggregation  $\delta_4$ :** for  $n_j \geq 2$  we obtain:

$$\begin{aligned} \delta_4(C_j) &= \frac{1}{2n_j} \sum_{\substack{x, x' \in C_j \\ x \neq x'}} d(x, x') = \frac{1}{2n_j} \left\{ \sum_{\substack{x, x' \in C_j \setminus \{x_i\} \\ x \neq x'}} d(x, x') + \sum_{x \in C_j} d(x, x_i) + \sum_{x \in C_j} d(x_i, x) \right\} \\ &= \frac{1}{2n_j} \left\{ 2(n_j - 1) \delta_4(C_j \setminus \{x_i\}) + 2 \sum_{x \in C_j} d(x, x_i) \right\} \\ &= \frac{n_j - 1}{n_j} \delta_4(C_j \setminus \{x_i\}) + \frac{1}{n_j} \sum_{x \in C_j} d(x, x_i). \end{aligned}$$

Result for  $\delta_4(C_j \setminus \{x_i\})$  is easily deduced.

**Aggregation  $\delta_5$ :** for  $n_j \geq 2$  we obtain:

$$\begin{aligned} \delta_5(C_j) &= \frac{1}{n_j(n_j - 1)} \sum_{\substack{x, x' \in C_j \setminus \{x_i\} \\ x \neq x'}} d(x, x') = \frac{1}{n_j(n_j - 1)} \left\{ \sum_{\substack{x, x' \in C_j \setminus \{x_i\} \\ x \neq x'}} d(x, x') + 2 \sum_{x \in C_j} d(x, x_i) \right\} \\ &= \frac{n_j - 2}{n_j} \delta_5(C_j \setminus \{x_i\}) + \frac{2}{n_j(n_j - 1)} \sum_{x \in C_j} d(x, x_i). \end{aligned}$$

Result for  $\delta_5(C_j \setminus \{x_i\})$  is then deduced.

**Aggregation  $\delta_6$ :** for  $n_j \geq 2$  we obtain:

$$\begin{aligned}\delta_6(C_j) &= \frac{1}{n_j(n_j-1)} \sum_{\substack{x, x' \in C_j \\ x \neq x'}} [d(x, x') - \mu(C_j)]^2 \\ &= \frac{1}{n_j(n_j-1)} \left\{ \sum_{\substack{x, x' \in C_j \setminus \{x_i\} \\ x \neq x'}} [d(x, x') - \mu(C_j)]^2 + 2 \sum_{x \in C_j \setminus \{x_i\}} [d(x, x_i) - \mu(C_j)]^2 \right\}\end{aligned}$$

The last term is decomposed using the Huygens lemma 4:

$$\sum_{x \in C_j \setminus \{x_i\}} [d(x, x_i) - \mu(C_j)]^2 = (n_j - 1)(n_j - 2) \{ \delta_6(C_j \setminus \{x_i\}) + [\mu(C_j) - \mu(C_j \setminus \{x_i\})]^2 \}$$

By substitution in the preceding formula, the formula for  $\delta_6(C_j \setminus \{x_i\})$  is deduced. ■

#### 4. METAHEURISTIC OF OPTIMIZATION

In Trejos **et al.** (1998) and Piza **et al.** (1999) we study the application of general metaheuristics to the partitioning problem, such as the simulated annealing, tabu search and genetic algorithm, in an Euclidean context. Results are significantly better than those of usual k-means or Ward methods.

##### 4.1. Simulated annealing

We begin choosing an initial random partition. For each "temperature" parameter  $t_m$ , we iterate with the following procedure.

At each step, we choose at random one object, say  $x_i$ . We also choose at random the index  $\ell$  of the new group to which the object  $x_i$  could be transferred. The transfer is actually made with probability  $\min \{1, e^{-\Delta W / t_m}\}$  (Metropolis Rule), where  $\Delta W$  is the change produced in the objective function  $W(P)$ .

After some iterations, we change the temperature parameter  $t_{m+1} < t_m$  (cooling the system) and repeat the transferring process, until a stop criteria is reached.

The cooling schedule used is:

1. **Initial temperature:**  $t_0$  is calculated in such a way that, at the beginning, the approximate probability of accepting new "bad partitions" (those that increase  $W(P)$ ), is about  $\chi \times 100$  %. This is done choosing  $t_0 := \Delta W_{\text{prom}}^+ / \ln(1/\chi)$ , where  $\Delta W_{\text{prom}}^+$  is the average of the change in the objective function  $W(P)$ , for partitions worst than the initial partition. We use  $\chi = 0.7$  with success.
2. **Cooling the temperature:** we calculate  $t_{k+1} = \lambda t_k$ , where  $\lambda = 0.92$  or another constant in  $[0.9, 1)$ .
3. **Large of iterative transferring process for each temperature  $t_k$ :** we use the homogeneous approach, in which the maximum large of the Markov chain is  $n_{\text{over}}$  steps. However, if  $n_{\text{limit}}$  new "bad partitions" were already accepted, then the temperature process stops. We use  $n_{\text{over}} = \min(100n^2(k-1), 20000)$  and  $n_{\text{limit}} = \min(10n^2(k-1), 4000)$
4. **Final temperature:** the algorithm stops at temperature  $t_{n_{\text{final}}}$ . However, we stop the algorithm if in the last  $n_{\text{cad}}$  temperature values no transfer was made. We use  $n_{\text{final}} = 150$  and  $n_{\text{cad}} = 3$  with success.

In theory, the simulated annealing algorithm converges asymptotically to an optimal solution of the problem, with probability 1.



## 4.2. Tabu search

A description of tabu search can be found in Murillo (in press). For the application of tabu search, a state is defined as a partition  $P$  and the neighbourhood is the set of all partitions  $P'$  defined by possible transfers  $C_j \xrightarrow{i} C_\ell$ . Object  $i$  and class  $\ell$  are chosen such that  $\Delta W$  in minimum, according to the tabu list handling (see for example Murillo (in press) or Trejos **et al.** (1998)). The indicator of the class of object  $i$  that is transferred, enters in the tabu list. Then, tabu list forbids  $i$  to be again with the same objects together in a class (at least until the indicator remains in the tabu list). The neighbourhood of a partition  $P$  has length  $n(k - 1)$ . If this number is large, tabu search spends too much time for generating the neighbourhood. In these cases, we use a sample of the neighbourhood, choosing at random some objects and come classes for making these transfers. This procedure works fine in the Pejibaye data set that will be presented later.

## 5. RESULTS

We present the results of our simulated annealing and tabu search methods on two data sets of objects described by binary variables. We computed three dissimilarities between the objects. For two objects  $i, j$ , these dissimilarities -among others- are based on the definition of:  $a_{ij}$  the number of attributes simultaneously present in  $x_i$ , and  $x_j$ ,  $b_{ij}$  the number of attributes present in  $x_j$  and  $n_i$  the number of attributes present in  $x_i$ . The indexes are:

a. Jaccard (1901):

$$d_1(x_i, x_j) = \begin{cases} 1 - \frac{a_{ij}}{a_{ij} + b_{ij} + c_{ij}} & \text{if } a_{ij} + b_{ij} + c_{ij} \neq 0 \\ 1 & \text{if } a_{ij} + b_{ij} + c_{ij} = 0 \end{cases}$$

b. Czekanowski (1913), Dice (1945), Sorensen (1948):

$$d_2(x_i, x_j) = \begin{cases} 1 - \frac{2a_{ij}}{n_i + n_j} & \text{if } n_i + n_j \neq 0 \\ 1 & \text{if } n_i + n_j = 0 \end{cases}$$

c. Russel y Rao (1940):

$$d_3(x_i, x_j) = 1 - \frac{a_{ij}}{p}.$$

### 5.1 Fictitious data

The fictitious data set is presented in Table 1; 20 objects are described by 6 binary variables and it is clear that there are 4 natural clusters:  $\{1, 2, 3, 4, 5\}$ ,  $\{6, 7, 8, 9, 10\}$ ,  $\{11, 12, 13, 14, 15\}$  and  $\{16, 17, 18, 19, 20\}$ .

We used the six aggregation indexes defined above. Tabu search runned 100 iterations with a tabu list of length 15; parameters for simulated annealing are described in the preceding section. Results for  $\delta_1$  and  $\delta_2$  are not interesting.

With simulated annealing and tabu search we obtained the same solutions for  $\delta_3$  and  $\delta_4$ , which is the natural partition; criteria  $W$  for  $\delta_3$  are 10 and 20 for  $d_1, d_2$  and  $d_3$ , respectively, and for  $\delta_4$  they are 2, 2 and 4. This natural partition was obtained in all runs of both methods.

For  $\delta_5$  and  $\delta_6$  we also obtained solutions that reach the global optimum of  $W$ , however these solutions are not interesting. For example, for  $\delta_5$  and  $d_1$  a solution is  $\{11, 12, 13, 14, 15\}$ ,  $\{18\}$ ,  $\{20\}$  and the remaining 13 objects in another class, with  $W = 0.5833$ . Aggregation  $\delta_5$  has a tendency to make singleton classes and to fill a class with many objects. On the other hand, a solution obtained for  $\delta_6$  and  $d_2$  is, for example,  $\{10, 16, 17, 18, 19, 20\}$ ,  $\{11, 12, 13, 14, 15\}$ ,  $\{1, 2, 3, 4, 5\}$ ,  $\{6, 7, 8, 9\}$  which missclassifies object 10, but for this aggregation this missclassification has no effect since the partition is optimal and the criterion is  $W = 0$ , as in the natural partition.

**Table 1.** The fictions binary data.

Object	
1	1 1 1 1 1 1
2	1 1 1 1 1 1
3	1 1 1 1 1 1
4	1 1 1 1 1 1
5	1 1 1 1 1 1
6	1 1 1 0 0 0
7	1 1 1 0 0 0
8	1 1 1 0 0 0
9	1 1 1 0 0 0
10	1 1 1 0 0 0
11	0 0 0 1 1 1
12	0 0 0 1 1 1
13	0 0 0 1 1 1
14	0 0 0 1 1 1
15	0 0 0 1 1 1
16	0 0 0 0 0 0
17	0 0 0 0 0 0
18	0 0 0 0 0 0
19	0 0 0 0 0 0
20	0 0 0 0 0 0

In all cases, both methods found the global optimum solution in few seconds, even if for some aggregation indexes ( $\delta_1$ ,  $\delta_2$ ,  $\delta_5$ ,  $\delta_6$ ) this solution is not the natural one.

## 5.2. Pejibaye data

The pejibaye (*Bactris gasipaes*) is a palm of the American humid tropic of great economical importance for this zone. We analized the "genetic trace" of 6 different populations of pejibaye coming from Brazil, Perú, Bolivia, Colombia, Panama and Costa Rica, deduce form a phytogenetic study made at the University of Costa Rica. The data set has 87 objects (genomic polymorphic fragments of pejibaye's plants) described by 60 binary variables (DNA genetic trace). The data are shown in Table 2.

We applied the simulated annealing and tabu search methods, using dissimilarities  $d_1, d_2, d_3$  and aggregations  $\delta_1, \dots, \delta_6$ . Tabu search was applied with 200 iterations and we sampled 20 % of the neighbours for deciding where to go in each iteration.

Solutions for  $\delta_1$  and  $\delta_2$  are not stable and may change on the runs; they are not reported here. Solutions for  $\delta_3$  and  $\delta_4$  are presented in Table 3. The partitions for the three dissimilarity indexes were always the same, for all runs with both methods.

**Table 2.** The pejibaye data set.

1:	111101100000011100000011011011110001110110111000100011100
2:	1101011000110111100000010110111101011101100111000100011100
3:	1001011000100111000000010110111101011101100111000100000000
4:	11100110001001111100011111010111001011001110011001100010000
5:	110101100001011100000000111011110101110110111000100011100
6:	1101011000000101000000001110111101011101101110011000001100
7:	0101011000100111000000101000101101101101100101001000001100
8:	1100011000100111010000101110011110010110110101001100010100
9:	110001100001010111000011100010111011110110111100110000100
10:	111001100001011100000001111011110101110110111001100001100
11:	0100011000110101100000001100011111010110110111001100011100
12:	0100011000010101110000011000101101011101101111001100001100
13:	110101100001010100000011111010111001110110111001100000100
14:	1101011000100111100000110111010110111110110111000100000100
15:	110111100011011100000010010101111101011011110001000001100
16:	11011100000010000100000111011100000011000000011111010010
17:	10101110000000110110000011010011011000111000000011111000010
18:	1101111100000100100001001101110110001110000000111010100
19:	0101000000000100100001001100110010000110000000110101100110
20:	11101100101001010100000100110011010011110000001110101100011
21:	11101110100001010100000100100111010111000001001111100111
22:	11011001001011100100001001001110101011000000011111101000
23:	0110011010110111001000010011001101010011000010011110001110
24:	01100111100001010010000001001111001011110000001010110001110
25:	1101100000000111000000011000001100101101101000011101101110
26:	11011000000101110000000000110111000011010010000011101110110
27:	11100100101101110000000000100111001011010000000011101100010
28:	111001101011010101000001111100110100001110000000110110000011
29:	011101101011010101000001101000110110001110000100110110010010
30:	01110110100000110000000100010111011011010000010111111010010
31:	00101101001011110000010110000011000000000101111100010110101
32:	0010110100101111100000011000001100001111111111100010110101
33:	001011000010111101000011000001100000110010111110001011101
34:	0010110000101011000010110000011000001100101111100100101000
35:	0010110100001111111001011000001100000110011111010010000000
36:	001011000010111000000101100001110010001001101110101110011100
37:	0010010100101110000001011000011100101010000010101111011101
38:	0010010100001111100001011000001100101110010010111101111001
39:	0010110100101111000010110000011001011100110101111001111011
40:	00101101001011111000000110000011000111100100111000000111111
41:	001011010010100100000000110000111000111001001110100100111101
42:	00101101001011001000000110000111001111001001110101110001000
43:	001001010010111000000110000011001000100100111111110011000
44:	0011010001011100010001010100010001110111010110100000010000
45:	0000000001111110010110111101110000000110010110110000000000
46:	01110100110111000001010111000111000111000100010000000000
47:	0010000001111110010001110110010011000111010100000000000000
48:	000001001110110000000110001000000110001110101100010000000000
49:	0011000001111100000001000101111001110011100011100000000000
50:	0000010011110011010111001100101000110011101011110000010000
51:	01100000011100010000010101100010011101111010111100000010000
52:	000101000101000101001001001000100011011111111110000000001
53:	0010000001000011010001101011011000101011110111111000000110
54:	00100000011011010001111101001000010100110111101110000000000
55:	000101000111110100000011011010000101001101010000000000000
56:	011000000111100000000101111100000100110101010000000010101
57:	00110000010111110001110011110000000100110111101100000011000
58:	000001011000000111000000001010100110011010111000000001010
59:	01100100100000011100000000110110011110011010111000000000010
60:	000011001000000111000000111110001010111101110000000000010
61:	0000010010000001110000000011000000001100110100110101100000010
62:	0110010110000001110000000110110001011011010111000000000010
63:	00000101100000010100000010011100010111110100111000000000010
64:	00101100100000111100000000101100111111101001110001000000011
65:	00000100100000111100000000111001011000110100110000100000011
66:	001001011000001111000000001011101000110110100110100000000110
67:	00100101100000110000000000001110110111011101001101100000110
68:	001001011000001111000000011011010111011010111000100001110
69:	00100101100000111000000000110110101110011010111000000001110
70:	00101100100000111000000000001100110101110100110110100001110
71:	001011001000001110000000001111000001001101001100000000001110
72:	0010110010000011100010000010111000001101101001100000000001110
73:	00100100010111011000000110100000001011110111011110011000100
74:	00100100000000111000000110000010000111110100011010011000100
75:	00000100110011000000000110001000100011011100111000000001000
76:	001001000011110010000001001110010000101010111000001000110
77:	001001000100110000000000100010001000010101001111000001000110
78:	001001001000111101000001100010001100110101110011100000000110
79:	00100100000110100000001110010001100110101110011011001100000
80:	000001000000000101000001100011100100111101010111001001100110
81:	001001001000110101000001100011000100000101011110001001000100
82:	00100100000011110100000110111100010001101000110001001000101
83:	001001011010111100000000110011001000010101000111000101100110
84:	00100100001010100000000110011001000011101011111010110110110
85:	00100100001010010000000010100100010110010101100101010001001
86:	001001001000101000000000100011001101010101100111001101001101
87:	001001000000101000000001100011001000100101001110000101001101

Partition obtained with  $\delta_4$  corresponds to the six countries of the peji-baye palms. In the sense, it is the natural and optimal solution. The difference between solutions for  $\delta_3$  and  $\delta_4$ , is that object 74 is classified in class  $C_3$  when it is used the sum of dissimilarities and in class  $C_6$  when it is used the Späth aggregation. One may think that  $\delta_3$  makes a missclassification of object 74, however, the value of  $W$  is less with the obtained classification using  $\delta_3$  than with classifying object 74 in class  $C_6$ . That is, the solution found by our methods is better (for  $\delta_3$  aggregation) than the "geographical" one.

**Table 3:** Partitions for the peji-baye data using simulated annealing and tabu search.

$\delta_3$ (sum of dissimilarities)	$\delta_4$ (Späth aggregation)
$C_1 = \{1,2,3,4,5,6,7,8,9,10, 11,12,13,14,15\}$	$C_1 = \{1,2,3,4,5,6,7,8,9,10, 11,12,13,14,15\}$
$C_2 = \{16,17,18,19,20,21,22,23,24,25,26,27,28,29,30\}$	$C_2 = \{16,17,18,19,20,21,22,23, 24,25,26,27,28,29,30\}$
$C_3 = \{31,32,33,34,35,36,37,38,39,40,41,42,43,74\}$	$C_3 = \{31,32,33,34,35,36,37,38, 39,40,41,42,43\}$
$C_4 = \{44,45,46,47,48,49,50,51,52,53,54,55,56,57\}$	$C_4 = \{44,45,46,47,48,49,50,51, 52,53,54,55,56,57\}$
$C_5 = \{58,59,60,61,62,63,64,65, 66,67,68,69,70,71,72\}$	$C_5 = \{58,59,60,61,62,63,64,65, 66,67,68,69,70,71,72\}$
$C_6 = \{73,74,75,76,77,78,79,80,81, 82,83,84,85,86,87\}$	$C_6 = \{73,74,75,76,77,78,79,80, 81,82,83,84,85,86,87\}$

Values of the criterion for different aggregations and dissimilarities are shown in Table 4. These results are for simulated annealing, and they are equal for tabu search using  $\delta_3$  and  $\delta_4$ . For  $\delta_5$  and  $\delta_6$  they differ slightly in some runs.

**Table 4.** Criterion  $W$  for the peji-baye data using simulated annealing according to 4 aggregations and 3 dissimilarities.

Aggregation	Jaccard ( $d_1$ )	Dice et al ( $d_1$ )	Russel & Rao ( $d_3$ )
$\delta_3$	263.904 (stable)	174.24 (stable)	406.47(stable)
$\delta_4$	18.19 (stable)	12.016 (stable)	28.018 (stable)
$\delta_5$	0.637 (stable)	0.477 (stable)	0.768 (stable)
$\delta_6$	0.0115(non-stable)	0.012 (non-stable)	0.00296(non-stable)

## 6. CONCLUSIONES

This study of aggregation indexes for non-Euclidean data shows that some aggregation indexes not involving the notion of center can be used for clustering data described by binary variables. Results obtained for the sum of dissimilarities and Späth aggregations are better than for the rest of agregations. Simulated annealing and tabu search find usually global optimum solutions for the data sets considered. Further research is undertaken for comparing our approach to methods that use centers and hierarchical clustering, as well as an extension for the use of aggregation indexes on categorical data. Genetic algorithms can also be applied for these aggregation indexes, since they satisfy the monotonicity property, even if not always there exist a decomposition of total inertia as in the Euclidean case.

## REFERENCES

- ABDALLAH, H. and G. SAPORTA (1998): "Classification d'un ensemble de variables qualitatives", **Revue de Statistique. Appliquée** XLVI (4):5-26.
- ANDERBERG, M. R. (1973): **Cluster Analysis for Applications**, Academic Press, New York.
- BORG, I. and P.J.F. GROENEN (1997): **Modern Multidimensional Scaling**, Springer, New York.
- BOCK, H.H. (1974): **Automatische Klassifikation**, Vandenhoeck & Ruprecht, Göttingen.
- CAILLIEZ, F. and J.P. PAGES (1976): **Introduction à l'Analyse des Données**, SMASH, Paris.
- DIDAY, E. (1980): **Optimisation en Classification Automatique**, (2 vol.), INRIA, Le Chesnay.
- DIDAY, E.; J. LEMAIRE; J. POUGET and F. TESTU (1982): **Eléments d'Analyse des Données**, Dounod, Paris.

- ESPINOZA, J.L. and J. TREJOS. (1989): "Clasificación por particiones". **Revista de Ciencia y Tecnología** 8 (1-2): 129-154.
- JAJUGA, K. (1987): "A clustering method based on the  $L_1$ -norm", **Computational Statistics Data Analysis** 5: 357-371.
- MURILLO, A. (in press): "La búsqueda tabú en el análisis de conglomerados", to appear in **Investigación Operacional**.
- PIZA, E. (1987): "Clasificación automática jerárquica aglomerativa", **Revista de Ciencias Económicas** 7(1).
- PIZA, E.; A. MURILLO y J. TREJOS (1999): "Nuevas técnicas de particionamiento en clasificación automática", **Revista de Matemática: Teoría y Aplicaciones** 6(1): 51-66.
- SPÄH, H. (1985): **Cluster Dissection and Analysis. Theory, Fortran programs, Examples**, Ellis Horwood, Chichester.
- TREJOS, J.; A. MURILLO and E. PIZA (1998): "Global stochastic optimization techniques applied to partitioning", in: A. Rizzi, M. Vichi & H.-H. Bock (Eds.), **Advances in Data Science and Classification**, Springer, Berlin.
- TREJOS, J. and M. VILLALOBOS. (in press): "Application of tabu search in metric multi-dimensional scaling", to appear in **Investigación Operacional**.
- VILLALOBOS, M. and J. TREJOS. (in press): "Application of simulated annealing in metric multidimensional scaling", to appear in **Investigación Operacional**.