COMBINATORIAL OPTIMIZATION HEURISTICS IN PARTITIONING WITH NON EUCLIDEAN DISTANCES

Eduardo Piza[1], Javier Trejos[2], y Alex Murillo[3], 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 sobrecalentamiento 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, sobrecalentamiento simulado, inercia generalizada.

MSC: 90C27

1. INTRODUCTION

Usual methods of partitioning (Forgy, k-means, dynamical clusters transfers, Isodata, etc.) find locall optima of the inertia (or general variance) criterion since they are based on procedures of locall 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 $D = (d(x,x^1)_{nxn}$ be a dissimilarity matrix on a set of n objects $\Omega = \{x_1,x_2,....,x_n\}$.

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

[1]E-mail:epiza@cariari.ucr.ac.cr. Fax: +(506) 207 4397
[2]E-mail:jtrejos@cariari.ucr.ac.cr. Fax: +(506) 207 4397
[3]E-mail:murillo@cariari.ucr.ac.cr Fax: +(506) 556 7020

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

where P_k is the set of all partitions of Ω in k or less classes, and $\delta(C\ )$ is some aggregation measure. Objects in Ω 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 heard to find and may be an element without representative sense, as often occurs in practice.

For the clustering of Ω 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 Ω.
- To define a good aggregation index δ and solve the problem using an adapted heuristic for partitioning.

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

3. AGGREGATION INDEXES

A dissimilarity index d on Ω 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 Ω. We study the following agggregation indexes defined on 2^{Ω}: 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:

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

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 $P = (C_1,...,C_k) \in P_k^*$ and $P' = \left(C_1',...,C_{k+1}'\right) \in P_{k+1}^*$ be partition of Ω in k and k+1 non-empty classes, respectively. We say that the aggregation index δ on 2^Ω satisfies the monotonicty property, if for all instances of the data, we have

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

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 Ω in k non-empty classes, while P_k is the set of all partitions of Ω in k or less classes.

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

Proof: Let us consider the partition $\hat{P} = (C_1,...,C_k)$ 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 $W(Q) \le W(\hat{P})$: begining with $\hat{P}$ we can construct Q by transfering 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: $Q = (C_1 \backslash \{x_i\}, C_2,...,C_k ,\{x_i\})$. The reader can easily precise the way to choose $x_i \in \Omega$ (in each case it depends on the aggregation index $\delta_1,....,\delta_6$) such that $W(Q) \le W(\hat{P})$. ν

Theorem 2 (Single linkage partition). Let x_p and x_q be objects of Ω with minimal dissimilarity. Then, a solution to the optimization problem corresponding to the index δ_1 (single linkage), $\min_{P\in P_k} W(P) = \sum_{j=1}^{k} \delta_1(C_j)$, is obtained for the partition $P^* = \left(C_1^*,...,C_k^*\right)$ of Ω 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 $P = (C_1,.....,C_k)$ of Ω in k classes, we have

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

Then, it is enough to choose any k - 1 objects of Ω, different from x_p and x_q and to distribute them in the initial k - 1 classes C_j, constructing singletons. The partition $P^* = \left(C_1^*,...,C_k^*\right)$ 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 $\delta_1,.....\delta_6$, there exist a solution $P^* \in P_{opt}$ that has non empty classes. Moreover, the aggregation indexes δ_3 (sum of dissimilarities) and δ_4 (Späth aggregation), all partitions $P \in P_{opt}$ have non empty classes.

Proof: The theorem holds for aggregation δ_1, as it was already proved. For the other aggregations δ_r, con $r \in \{2,...,6\}$, let $P = (C_1,...C_k) \in P_{opt}$ and suppose that P has an empty class. Let us see how to "fill" that class by transfering an object x_i, chosen in any of the other non unitary classes, say from class C, in such a way that inequality $\delta_r(C\backslash\{x_i\}) \leq \delta_r(C)$ 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 $\delta_r(C\backslash\{x_i\})$.

(a) Aggregation δ_2: it can be chosen any $x_i \in C$, since it is always satisfied $\max\{d(x,x'): x,x' \in C\backslash\{x_i\}\} \leq \max\{d(x,x'): x,x' \in C\}$, for any $x_i \in C$.

(b) Aggregation δ_3: any choice of $x_i \in C$ is useful, since $\delta_3(C\backslash\{x_i\}) = \delta_3(C) - \sum_{x \in C} d(x, x_i) < \delta_3(C)$. Remark that the inequality is strict, hence the partition obtained by the transfer of x_i always improves the criterion.

(c) Aggregation δ_4: solving the inequality $\delta_4(C\backslash\{x\}) \leq \delta_4(C)$, we obtain the equivalent inequality $\delta_4(C\backslash\{x_i\}) \leq \sum_{x \in C} d(x, x_i)$ 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 δ_5: solving the inequality $\delta_5(C\backslash\{x_i\}) \leq \delta_5(C)$, 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 δ_6: the inequality $\delta_6(C\backslash\{x_i\} \leq \delta_6(C)$ is satisfied al least when we choose any object $x_i \in C$ that maximizes the quantity $\frac{1}{|C|(|C|-1)} \sum_{x \neq x' \in C} \left[d(x, x') - \mu(C)\right]^2$. Also in this case the quantities $\delta_6(C\backslash\{x_i\})$ and $\delta_6(C)$ are equal when all the dissimilarities between objects of C are equal.

This reasoning is repeated by transfering 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 δ_3 and δ_4 are used. It can be also remarked that aggregations δ_5 and δ_6 have a tendency to produce optimal partitions without empty classes, with the pointed out exceptions.

Lemma 4 (Huygens decomposition using δ_6) For any class $C \subseteq \Omega$ and any real number β, 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

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

$$+ 2 \sum_{\substack{x,x' \in C \\ x \neq x'}} [d(x,x') - \mu(C)][\mu(C) - \beta]$$

$$= |C|(|C| - 1) \{\delta_6(C) + [\mu(C) - \beta]^2\}$$

$$+ 2[\mu(C) - \beta] \sum_{\substack{x,x' \in C \\ x \neq x'}} [d(x,x') - \mu(C)].$$

The last term is clearly mull, 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 $P = (C_1, \ldots, C_k)$ and $\tilde{P} = (\tilde{C}_1, \cdots, \tilde{C}_k)$ be two partitions on Ω, such that object i is transfered 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 $\delta_1 \ldots \delta_6$.

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

$$\delta_1(C_\ell \cup \{x_i\}) = \min(\delta_1(C_\ell), \min\{d(x,x_i) : x \in C_\ell\}.$$

$$\delta_2(C_\ell \cup \{x_i\}) = \max(\delta_2(C_\ell), \max\{d(x,x_i) : x \in C_\ell\}.$$

$$\delta_3(C_\ell \cup \{x_i\}) = \delta_3(C_\ell) + \sum_{x \in C_\ell} d(x,x_i).$$

$$\delta_4(C_\ell \cup \{x_i\}) = \frac{n_\ell}{n_\ell + 1} \delta_4(C_\ell) + \frac{1}{n_\ell + 1} \sum_{x \in C_\ell} d(x,x_i).$$

$$\delta_5(C_\ell \cup \{x_i\}) = \frac{n_\ell - 1}{n_\ell + 1} \delta_5(C_\ell) + \frac{2}{n_\ell(n_\ell + 1)} \sum_{x \varepsilon C_\ell} d(x,x_i).$$

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

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 δ_1, δ_2 and δ_3 are elementary. We will deal only with the rest:

Aggregation δ_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 δ_5:

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

Aggregation δ_6: we use Huygens lemma 4:

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

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

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

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

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

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

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 δ_1 and δ_2 do not have a recursive formula for the computation of $\delta(C_j \setminus \{x_i\})$.

Proof: The formula for δ_3 is elementary. We will deal only with the rest.

Aggregation δ_4: for $n_j \geq 2$ we obtain:

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

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

Aggregation δ_5: for $n_j \geq 2$ we obtain:

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

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

Aggregation δ_6: for $n_j \geq 2$ we obtain:

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

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

The last term is descomposed using the Huygens lemma 4:

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

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 ℓ of the new group to which the object x_i could be transfered. The transfer is actually made with probability $\min\left\{1, e^{-\Delta W / t_m}\right\}$ (Metropolis Rule), where ΔW is the change produced in the objetive function W(P).

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

The cooling schedule used is:

1. **Initial temperature:** t_o is calculated in such a way that, at the begining, the approximate probability of accepting new "bad partitions" (those that increase W(P)), is about $\chi \times 100$ %. This is done choosing $t_o := \Delta W^+_{prom} / \ln(1/\chi)$, where ΔW^+_{prom} is the average of the change in the objetive 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 transfering process for each temperature t_k**: we use the homogeneous approach, in which the maximum large of the Markov chain is n_{over} steps. However, if n_{limit} new "bad partitions" were already accepted, then the temperature process stops. We use $n_{over} = \min(100n^2(k-1),20000)$ and $n_{limit} = \min(10n^2(k-1),4000)$

4. **Final temperature**: the algorithm stops at temperature $t_{n_{final}}$. However, we stop the algorithm if in the last n_{cad} temperature values no transfer was made. We use $n_{final} = 150$ and $n_{cad} = 3$ 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 ℓ are chosen such that ΔW in minimum, acording 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 transfered, 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 disimilarities between the objects. For two objects i,j, these disimilarities -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):

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

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

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

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 δ_1 and δ_2 are not interesting.

With simulated annealing and tabu search we obtained the same solutions for δ_3 and δ_4,which is the natural partition; criteria W for δ_3 are 10 and 20 for d_1,d_2 and d_3, respectively, and for δ_4 they are 2, 2 and 4. This natural partition was obtained in all runs of both methods.

For δ_5 and δ_6 we also obtained solutions that reach the global optimum of W, however these solutions are not interesting. For example, for δ_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 δ_5 has a tendency to make singleton classes and to fill a class with many objects. On the other hand, a solution obtained for δ_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 efect since the partition is optimal and the criterion is W = 0, as in the natural partition.

Table 1. The fictitions 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 (δ_1, δ_2, δ_5, δ_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 analizad 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 δ_1,....,δ_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 δ_1 and δ_2 are not stable and may change on the runs; they are not reported here. Solutions for δ_3 and δ_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:	111101100000011100000011011101111000111101101111000100011100
2:	110101100011011110000001011101111010111101100111000100011100
3:	100101100010011100000001011101111010111101100111000100000000
4:	111001100010011111100011111101011100101101100111001100001000
5:	110101100001011100000000111101111010111101110111000100011100
6:	110101100000010100000000111101111010111101110111001000011100
7:	010101100010011100000010110001011011101101100101001000001100
8:	110001100010011101000010111001111100101101110101001100010100
9:	110001100001010111000011110001011110111101101111001100000100
10:	111001100001011100000000111101111010111101101111001100001100
11:	010001100011010110000000110001111110101101101111001100011100
12:	010001100001010111000001110001011010111101101111001100001100
13:	110101100001010100000001111110101110011110110111100110000010
14:	110101100010011110000011011101011011111101101111000100000100
15:	110111100011011100000001001010111111010110110111100010001110
16:	111011100000010000100000111101110000001110000000111111010010
17:	101011100000001101100000110100110110001110000000111111000010
18:	111011111000001100100001001101110110001110000000111101010100
19:	011010000000001100100001001100110010000110000000110101100110
20:	111011001010010101000001001100110100111110000000111010110001
21:	111011101000010101000001001001110110111100000100111111100111
22:	111011001001011100100001001001110110101100000001111111101000
23:	011001101011011100100001001100110110100110000100111110001110
24:	011001111000010100100000010011110010111110000000101011000111
25:	111011000000011100000001100000110010110110100001110111011110
26:	110110000001011100000000000110111000011010010000011110111011
27:	111001001011011100000000000100111001011010000000011110110001
28:	111001101011010101000001111100110100001110000000110110000011
29:	011101101011010101000001101000110110001110000100110110010010
30:	011101101000001100000000100010111011011010000001011111111010
31:	001011010010111100000101100000110000000001011111100010110101
32:	001011010010111110000001100000110000111111111111100010110101
33:	001011000010111110100001100000110000011001011111100010111101
34:	001011000010101110000101100000110000011001101111100100101000
35:	001011010000111111100101100000110000011001111110100100000000
36:	001011000010111000000101100001110010001001101110101110011100
37:	001001010010111000000101100001110010101000001010111111011101
38:	001001010000111111000010110000011001011100100101111101111100
39:	001011010010111111000010110000011001011100110101111100111101
40:	001011010010111111000000110000011000111100100111000000111111
41:	001011010010100100000000110000111000111100100111010010011110
42:	001011010010110010000000110000111001111100100111010111000100
43:	001001010010101110000000110000011001000100100111111111001100
44:	001101000101110001000101101000100011101110101101100000010000
45:	000000000111111100101101111011100000001100101101100000000000
46:	011101001101110000010101101110001111001110001000100000000000
47:	001000000111111001000111101100100110001110101000000000000000
48:	000001001110110000000110001000000110001110101100010000000000
49:	001100000111110000000010001011110011100111000111001000000000
50:	000001001111001101011100110010100011001110101111100000010000
51:	011000000111000100000101011000100111011110101111100000010000
52:	000101000101000101001001001000100011011110111111111000000001
53:	001000000100001101000110101101100010101111011111111000000110
54:	001000000110110100011111010010000101001101111011100000000000
55:	000101000111111010000001101101000010100110110101000000000000
56:	011000000111110000000001011111000000100110110101000000010101
57:	001100000101111100011100111100000001001101111011100000011000
58:	000001011000000111000000000101101001100110101110000000001010
59:	011001001000000111000000001101100111100110101110000000000010
60:	000011001000000011100000011111100010101111101110000000000010
61:	000001001000000011100000000101110011010011010110000010000001
62:	011001011000000111000000011011100010110110101111000000000010
63:	000001011000000101000000001001110001011111010011100000000001
64:	001011001000000111100000000010110011111111010011100010000001
65:	000001001000000111100000000111100101100011010011000010000001
66:	001001011000000111100000000101110100011011010011010000000011
67:	001001011000000110000000000011101101110111010011011010000111
68:	001001011000000111100000001101110101110110101111000100001110
69:	001001011000000111000000001101110101100110101110000000001110
70:	001011001000000111000000000000110011010111010011011010000111
71:	001011001000000111000000000111110000010011010011000000000111
72:	001011001000000111000100000101110000011011010011000000000111
73:	001001000101110110000001101000000010111101110111110011000100
74:	001001000000000111000000110000010000111110100011101001100010
75:	000001001100110000000000110001000100011011110011100000100100
76:	001001000011111001000000010011100100001010101011100000100011
77:	001001000100110000000000010001000100001010100111100000100011
78:	001001001000111101000000110001000110011010111001110000100011
79:	001001000000110100000000111001000110011010111001101100110000
80:	000001000000000101000000110001110010011110101011100100110011
81:	001001001000110101000000110001100010000101101111000100100010
82:	001001000000111101000000110111100010001110100011000100100010
83:	001001011010111100000000011001100100001010100011100010110011
84:	001001000010110100000000011001100100001110110111111010110011
85:	001001000010100100000000010100100010110010101011001010100010
86:	001001001000101000000000010001100110101010110011100110100110
87:	001001000000101000000000110001100100010010100111000010100110

Partition obtained with δ_4 corresponds to the six countries of the pejibaye palms. In the sense, it is the natural and optimal solution. The difference between solutions for δ_3 and δ_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 δ_3 makes a missclassification of object 74, however, the value of W is less with the obtained classification using δ_3 than with classifying object 74 in class C_6. That is, the solution found by our methods is better (for δ_3 aggregation) than the "geographical" one.

Table 3: Partitions for the pejibaye data using simulated annealing and tabu search.

δ_3 (sum of dissimilarities)	δ_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 = {73,74,75,76,77,78,78,80,81, 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 δ_3 and δ_4. For δ_5 and δ_6 they differ slightly in some runs.

Table 4. Criterion W for the pejibaye data using simulated annealing according to 4 aggregations and 3 dissimilarities.

Aggregation	Jaccard (d_1)	Dice et al (d_1)	Russel & Rao (d_3)
δ_3	263.904 (stable)	174.24 (stable)	406.47(stable)
δ_4	18.19 (stable)	12.016 (stable)	28.018 (stable)
δ_5	0.637 (stable)	0.477 (stable)	0.768 (stable)
δ_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 anneanling 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, París.

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, París.

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**.