Journal of Classification 14:311-325 (1997)

Recognition of Robinsonian Dissimilarities

Victor Chepoi
Université d'Aix Marseille II

Bernard Fichet
Université d'Aix Marseille II

Abstract: We present an $O(n^3)$-time, $O(n^2)$-space algorithm to test whether a dissimilarity d on an n-object set X is Robinsonian, i.e., X admits an ordering such that $i \leq j \leq k$ implies that $d(x_i,x_k) \geq \max\{d(x_i,x_j), d(x_j,x_k)\}$.

Résumé: Nous présentons un algorithme de complexité $O(n^3)$ pour le temps et $O(n^2)$ pour l'espace mémoire, afin de tester si une dissimilarité d sur un ensemble X de n objets est de Robinson, i.e., si X admet un ordre tel que $i \leq j \leq k$ entraîne $d(x_i,x_k) \geq \max\{d(x_i,x_j), d(x_j,x_k)\}$.

Keywords: Robinsonian dissimilarities; Order compatible with a dissimilarity; Divide-and-conquer algorithm.

1. Introduction

A major issue in classification and dissimilarity analysis is to visualize simple geometrical and relational structures between objects. Necessary for such an analysis is a dissimilarity measure d on a set X of objects, which is either observed directly or computed from a data matrix. Then d is a symmetric function from X^2 to the nonnegative real numbers and vanishing on

---

The authors thank the editor and the referees for their suggestions and helpful comments.

Authors' Addresses: Victor Chepoi and Bernard Fichet, Laboratoire de Biomathématiques, Faculté de Médecine, Université d'Aix Marseille II, 27 Bd Jean Moulin 13385 Marseille Cedex 5 France. The first author is on leave from the Universitatea de stat din Moldova, Chisinau.

the diagonal. Hierarchical structures (or "dendrograms") are basic tools in such a visual display. A dissimilarity is in perfect agreement with a dendrogram if and only if it satisfies the ultrametric inequality $d(x,y) \leq \max \{d(x,z),d(y,z)\}$ for all $x, y, z \in X$. Moreover, the left-to-right order $\langle$ of objects in the dendrogram is compatible with d, i.e., if $d(x,y) < \max \{d(x,z),d(y,z)$ and $x \langle y$, then either $z \langle x$ or $y \langle z$. Such an ordering is still another (one-dimensional) way to represent ultrametrics (Brossier 1980), although the class of dissimilarities admitting compatible orders is much larger. This kind of orderings appeared in Robinson (1951), concerning seriation problems in archaeology.

A dissimilarity d and a (total) order $\langle$ on a set X are said to be *compatible* (or *admissible* sensu Mirkin and Rodin 1984, p. 62) if $x \langle y \langle z$ implies $d(x,z) \geq \max \{d(x,y),d(y,z)\}$. A dissimilarity d is said to be *Robinsonian* if it admits a compatible order. Then the dissimilarity matrix $D = (d(x,y))_{x,y \in X}$ has elements which do not decrease when moving away from the main diagonal along any row or column. Such a matrix is called Robinsonian (cf. Brossier 1980; Critchley and Fichet 1994; Diday 1983, 1986; Fichet 1986; Hubert 1974) (or linear in the terminology of Mirkin and Rodin 1984, p. 39). Finally, recall also a stronger kind of Robinsonian dissimilarity measure. A dissimilarity d on X is called *strongly-Robinsonian* (cf. Fichet 1986; Durand and Fichet 1988) if it admits an order $\langle$ compatible with d, such that $x \langle y \langle z \langle t$ and $d(x,t) > \max\{d(x,z),d(t,y)\}$ imply $d(y,z) < \min\{d(x,z),d(t,y)\}$. In this case the order $\langle$ is *strongly-compatible* with d. For the state of art in this field, the reader may consult the recent book of Mirkin (1996).

In addition to seriation problems in archaeology (Robinson 1951; Hubert 1974; Kendall 1969), Robinsonian dissimilarities play an important role in the analysis of DNA sequences (Mirkin and Rodin 1984, Chapter 1) and overlapping clustering (Bertrand and Diday 1985; Diday 1986; Durand and Fichet 1988; Fichet 1986). There are one-to-one correspondences between, on the one hand, Robinsonian dissimilarities and weakly indexed pseudo-hierarchies (alias pyramids) (Diday 1986), and, on the other hand, strongly-Robinsonian dissimilarities and strictly indexed pseudo-hierarchies (Fichet 1986). Finally, recall the basic relationship between Robinsonian matrices and interval hypergraphs established by Fulkerson and Gross (1965). For further information on applications and properties of Robinsonian dissimilarities, consult Batbedat (1990), Bertrand (1992), Critchley and Fichet (1994), and Chapter 1 of Mirkin and Rodin (1984). Hence, given a dissimilarity d on X, one can be interested in recognizing whether d is Robinsonian and in finding a compatible order if one exists.

The simplest approach to this problem uses the following property of Robinsonian dissimilarities: if $x_1, \ldots, x_n$ is an order of X compatible with d, then the chain $(x_1, \ldots, x_n)$ is a minimum spanning tree in (X,d). Therefore, it

suffices to find all minimum spanning trees that are chains and check if one of them gives an order compatible with d. Unfortunately, because (X,d) can have an exponential number of minimum spanning trees, it is unclear whether such an algorithm can be implemented in polynomial time. The same difficulties arise with the estimation of complexity of the algorithms presented in Batbedat (1990), Bertrand (1986), and Durand (1989). The polynomial-time algorithm, unique to our knowledge, is based on the simple fact that a dissimilarity d on X is Robinsonian if and only if the family $\mathbf{B}$ of all balls $B_r(v) = \{x \in X: d(v,x) \leq r\}$, $v \in X$ is an *interval hypergraph*. (Such an approach has been used, for example, in Mirkin and Rodin 1984, Chapter 1.) Thus, it is necessary to compute the ball-hypergraph $\mathbf{B}$ and check whether its incidence matrix can be reordered in such a way that every column has the consecutive one's property; see Golumbic (1980, Chapter 8) for more information about such matrices and their relationships with interval graphs and hypergraphs. The last problem can be solved in time proportional to the number of units of this matrix (i.e., the size of $\mathbf{B}$) by the PQ-tree algorithm of Booth and Lueker (1976). In the worst case, the hypergraph $\mathbf{B}$ has size $O(n^3)$ (one cannot know in advance which balls coincide), and, therefore, Robinsonian dissimilarities can be recognized in $O(n^3)$ time and with $O(n^3)$ space. We now present a direct (and probably simpler) algorithm for solving this problem, which needs $O(n^3)$ time and $O(n^2)$ space. With a few modifications we can recognize strongly-Robinsonian dissimilarities within the same time and space bounds.

2. Preliminaries

Our algorithm makes use of some simple geometrical and order-theoretical notions. Let d be a dissimilarity on a set X of n objects. We call (by a slight abuse of language) $d(x,y)$ the distance between the objects $x,y \in X$. For a subset $A \subseteq X$ let $\delta(A) = \max\{d(u,v): u,v \in A\}$ be the *diameter* of A. By $S_r(x)$ we denote the *sphere* of radius r and centered at $x \in X$, i.e., $S_r(x) = \{y \in X; d(x,y) = r\}$. For convenience, x does not belong to $S_0(x)$.

Let $\mathbf{R}$ be a *(linear) quasi-order* on X, i.e., a reflexive, transitive, and linear binary relation. It is common to write $x \leq_{\mathbf{R}} y$ to indicate that (x,y) is in $\mathbf{R}$. Any quasi-order $\mathbf{R}$ can be represented as an ordered partition $(B_1, \ldots, B_m)$, where $x \leq_{\mathbf{R}} y$ if and only if $x \in B_i$, $y \in B_j$ and $i \leq j$. Thus, $B_1, \ldots, B_m$ are the *blocks* (equivalence classes) of $\mathbf{R}$. By a *segment* $[B_i,B_j]$ $(i < j)$, is meant the union of the consecutive blocks $B_i, B_{i+1}, \ldots, B_j$. For two subsets of objects P and Q, write $P \leq_{\mathbf{R}} Q$ if $p \leq_{\mathbf{R}} q$ for any $p \in P$ and $q \in Q$. A quasi-order $\mathbf{R}_2$ *refines (extends)* a quasi-order $\mathbf{R}_1$ if every block of $\mathbf{R}_1$ is a segment of $\mathbf{R}_2$, or, equivalently, $x \leq_{\mathbf{R}_2} y$ implies $x \leq_{\mathbf{R}_1} y$. Note that this refinement relation on quasi-orders is nothing but the inclusion of binary

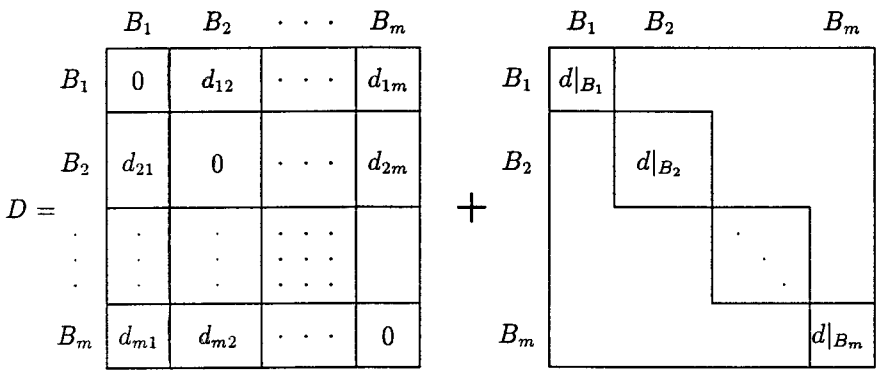

Figure 1. Decomposition of the matrix D given a quasi-order compatible with d.

relations considered as sets of ordered pairs. Henceforth, whenever possible we will reserve the notation $\langle$ for (total) orders.

The following is a standard example of a quasi-order of X. Pick an object $b \in X$ and define $x \leq_{\mathbf{R}_b} y$ if and only if $d(b,x) \leq d(b,y)$. The blocks of the quasi-order $\mathbf{R}_b$ are the spheres centered at b.

We will say that a quasi-order $\mathbf{R}$ of X with the blocks $B_1, \ldots, B_m$ is *compatible* with a dissimilarity d if $x \in B_i$, $y \in B_j$, $z \in B_k$ and $i \leq j \leq k$, $i \neq k$, implies

$$d(x,z) \geq \max \{d(x,y), d(y,z)\} .$$

This notion clearly generalizes the concept of a (total) order compatible with d. On the other hand, the quasi-order with one block is compatible with any dissimilarity. The following properties of quasi-orders $\mathbf{R}$ compatible with d are straightforward but instrumental:

(1) for any two distinct blocks B_i and B_j of $\mathbf{R}$, the distances $d(x,y)$ $(x \in B_i,\ y \in B_j)$ take one and the same value $d_{ij} = d_{ji}$ (see Figure 1).

(2) if $i < j < k$, then $d_{ik} \geq \max \{d_{ij}, d_{jk}\}$, i.e., the induced (total) order between the blocks of $\mathbf{R}$ is compatible with the dissimilarity

$$d^*(B_l, B_t) = \begin{cases} d_{lt}, & \text{if } l \neq t, \\ 0 & \text{otherwise} . \end{cases}$$

(3) the diameter $\delta(B_i)$ of any block B_i is not larger than d_{ii+1} and d_{i-1i}.

(4) d is Robinsonian if and only if its restriction $d \mid_{B_i}$ on any block of $\mathbf{R}$ is Robinsonian. Moreover, if $\mathbf{R}_i$ are orders/quasi-orders of B_i $(i = 1, \ldots, m)$ compatible with $d \mid_{B_i}$, then the following

order/quasi-order $\mathbf{R}^*$ of X is compatible with d and refines $\mathbf{R}$:

$x \leq_{\mathbf{R}^*} y$ if and only if either x and y belong to a common block B_i and $x \leq_{\mathbf{R}_i} y$, or $x \in B_i$, $y \in B_j$ and $i < j$.

3. Algorithm

We start with an outline of the algorithm for recognizing Robinsonian dissimilarities. The last property from the previous section and Figure 1 suggest that a divide-and-conquer paradigm with a simple merging step can be employed. The uniquely serious obstacle is the division step: we must find (if it exists) a quasi-order $\mathbf{R}$ compatible with d and having at least two blocks. We proceed in two steps. The approach begins by finding a quasi-order $\mathbf{R}_0$ with at least two (and at most six) blocks and fulfilling the following condition: *if the dissimilarity d is Robinsonian, then there is a (total) order of X compatible with d which refines* $\mathbf{R}_0$. Then, to construct the desired quasi-order $\mathbf{R}$ it is necessary to sweep the objects according to the quasi-order $\mathbf{R}_0$. For an object b under consideration, we refine the current quasi-order (initially $\mathbf{R}_0$) by intersecting its blocks not comprising b with the blocks of the quasi-order $\mathbf{R}_b$ (recall that they are the spheres centered at b). We recursively repeat this sweeping procedure inside some blocks of $\mathbf{R}_0$ until we arrive at a quasi-order where each block is entirely contained in only one sphere $S_r(b)$ for any object b outside this block. Denote the quasi-order thus obtained by $\mathbf{R}$. Note that any order of X compatible with d and which refines $\mathbf{R}_0$ will refine $\mathbf{R}$, too. Therefore, d is Robinsonian if and only if $\mathbf{R}$ is compatible with d. If so, continue the same two-level procedure for the blocks of $\mathbf{R}$ separately; otherwise return the negative answer.

In the following pages a rather detailed description of the algorithm is given. Besides the dissimilarity matrix D, we will need the matrix $D_{\leq}$ obtained from D by sorting the row elements in nondecreasing order. This task is accomplished in $O(n^2 \log n)$ time. Next, compute the diameter $\delta := \delta(X)$ of X and find an object p such that the sphere $S := S_\delta(p)$ contains the maximum number of objects. Select all potential centers of this sphere, i.e., all $c \in X$ with the property that $S_\delta(c) = S$. Denote the collection of all such c by C. Finally, put $L := X - (C \cup S)$. We record here some elementary properties of the sets C, S and L. First notice that S and C must be disjoint. Indeed, if there is an object $q \in S \cap C$, then $\delta > 0$ by assumption and $d(q,q) = \delta > 0$, which is impossible.

Lemma 1. *If* $L = \emptyset$, *then the quasi-order* $\mathbf{R}_0 = (S,C)$ *is compatible with d.*

Proof. The proof is immediate, because $d(x,z) = \delta \geq \max\{d(x,y),d(y,z)\}$ for any objects $x \in S$, $z \in C$ and $y \in S \cup C$. In this case set $\mathbf{R} := \mathbf{R}_0$, avoiding the refining step of the algorithm.

Lemma 2. *C is a segment of any order $\langle$ of X compatible with d.*

Proof. For a given compatible order, let c' and c'' be the respective minimum and maximum elements of C. We assert that the segment $[c', c''] = \{x \in X: c' \langle x \langle c''\}$ does not contain objects of S. Assume by way of contradiction that $q \in S \cap [c', c'']$. Since $d(c', q) = d(c'', q) = \delta$ and the order $\langle$ is compatible with d, we conclude that $d(c', c'') = \delta$. This conclusion implies $c', c'' \in C \cap S$, which is impossible. Thus, $[c', c''] \cap S = \varnothing$.

Now, pick arbitrary objects $c \in [c', c'']$ and $s \in S$. Let, say, $s \langle c'$. Then $d(c,s) \geq d(c', s) = \delta$ and hence $s \in S_\delta(c)$. From the choice of the sets C and S we deduce that $S_\delta(c) = S$, i.e., $c \in C$. ∎

For an order $\langle$ compatible with d, define

$$S_\langle^- = \{s \in S: s \langle C\}, \quad S_\langle^+ = \{s \in S: C \langle s\},$$

$$L_\langle^- = \{l \in L: l \langle C\}, \quad L_\langle^+ = \{l \in L: C \langle l\}.$$

By Lemma 2 we conclude that $S_\langle^- \cup S_\langle^+ = S$ and $L_\langle^- \cup L_\langle^+ = L$.

Lemma 3. *The sets $S_\langle^-$, $L_\langle^-$, $L_\langle^+$ and $S_\langle^+$ are segments of the order $\langle$. Moreover, $\langle$ refines the quasi-order with the blocks $(S_\langle^-, L_\langle^-, C, L_\langle^+, S_\langle^+)$.*

Proof. Pick arbitrary objects $s \in S_\langle^-$ and $x \langle s$. Since $d(x,c) \geq d(s,c) = \delta$ for any $c \in C$, we obtain $x \in S_\langle^-$. Similarly, if $s \in S_\langle^+$ and $s \langle x$, then $x \in S_\langle^+$. Hence, both $S_\langle^-$ and $S_\langle^+$ are boundary segments of $\langle$. This observation immediately implies that $L_\langle^-$ and $L_\langle^+$ are segments, too. In view of this observation and Lemma 2, the second assertion is evident. ∎

Next we will explain how to find such a partition without the knowledge of any order compatible with d. For this purpose construct a special *threshold graph* G with $L \cup S$ as a vertex-set and the following set E of edges: define $(x,y) \in E$ if and only if $d(x,y) < \delta$ and x and y do not belong simultaneously to the set L. Hence, L is an independent (stable) set of G. Applying the breadth-first search (see, for example, Golumbic 1980, p. 39) starting every time from L, we will find the connected components $K_1, \ldots, K_m$ of the graph G that share objects with L. Define $L_1 = K_1 \cap L, \ldots, L_m = K_m \cap L$ and $S_1 = K_1 \cap S, \ldots, S_m = K_m \cap S$. Since

any object in L is at distance less than δ to at least one object of S, necessarily all the sets $S_1, \ldots, S_m$ must be nonempty. Finally, put $S^{=} = S - (S_1 \cup \cdots \cup S_m)$. (In fact, one can compute directly all these sets without an explicit construction of the graph G, using just the matrices D and $D_{\leq}$).

Lemma 4. *Let $\langle$ be an order of X compatible with d. Then G has at most two connected components intersecting L. Moreover, for any connected component K_i of the graph G, the sets K_i, L_i, S_i and $K_i \cup C$ represent nonempty segments of $\langle$.*

Proof. Let a and b be the respective minimum and maximum elements of K_i. We will prove that $K_i = [a,b]$. Pick an arbitrary object $z \in [a,b]$ distinct from a and b. Necessarily, one can find two objects $x,y \in K_i$ adjacent in G and such that $x \langle z \langle y$. Indeed, when moving from a to b on a path of K_i, such a pair (x,y) must be encountered. At least one of x or y, say x, belongs to the set S. Since $\max\{d(x,z),d(z,y)\} \leq d(x,y) < \delta$, we conclude that $z \notin C$. Therefore, z is a vertex of G, and z either coincides with x or is adjacent to x. Hence, $z \in K_i$, i.e., K_i is a segment of $\langle$. Suppose without loss of generality that $K_i \langle C$. Then $S_i \subseteq S_{\langle}^{-}$ and $L_i \subseteq L_{\langle}^{-}$. By Lemma 3 and the simple fact that intersections of segments are segments too, we infer that S_i and L_i are segments. Now, we show that $L_i = L_{\langle}^{-}$. Pick an arbitrary object $u \in L_{\langle}^{-}$. The definition of the set C implies that $d(u,v) < \delta$ for at least one object $v \in S$. Since $d(u,t) \geq d(w,t) = \delta$ for all $w \in C$ and all $t \in S_{\langle}^{\dagger}$, we conclude that $v \in S_{\langle}^{-}$. If $v \in S_i$, then immediately $u \in K_i$. Otherwise, $v \langle a$, so that $d(a,u) \leq d(v,u) < \delta$, and again $u \in K_i$. In both cases $u \in L_i$, i.e., $L_i = L_{\langle}^{-}$. This shows that $K_i \cup C$ is a segment, and that G has at most two connected components. ■

Lemma 4 shows that the connected components K_1 and K_2 of the graph G together with the set C represent a "rigid part" of any order of X compatible with d.

Below we give a more formal description of the dividing strategy.

procedure divide (X)

Input: a dissimilarity d of X given by the matrices D and $D_{\leq}$;
Output: a quasi-order $\mathbf{R}_0$ of X with at most six blocks;

begin
compute the diameter $\delta := \delta(X)$ of X;
find an object p with the largest sphere $S := S_{\delta}(p)$;
compute $C := \{c \in X : S_{\delta}(c) = S\}$; $L := X - (S \cup C)$;

```
    if L = ∅ then return R0 = (S,C);
        else
            construct the threshold graph G;
            find the connected components K1 = L1 ∪ S1, . . . ,Km = Lm ∪ Sm
            of G which intersect the set L;
            if m ≥ 3 then return "d IS NOT ROBINSONIAN";
            else
                S⁻ := S1; L⁻ := L1; S⁺ := S2; L⁺ : =L2; S⁼ := S − (S⁻ ∪ S⁺);
                if m = 1 then return R0 = (S⁻ , L⁻ , C,S⁼);
                else return R0 = (S⁻ , L⁻ , C,L⁺ , S⁺ , S⁼);
end;
```

We summarize the previous results in the following theorem.

Theorem 1. *Any order of X compatible with d is a refinement of a quasi-order of the following type:*

(a) (S,C) *or* (C,S), *if* $L = \emptyset$;
(b) (S', S^-, L^-, C, S'') *or* (S', C, L^-, S^-, S''), *if* $L^- = L$;
(c) $(S', S^-, L^-, C, L^+, S^+, S'')$ *or* $(S', S^+, L^+, C, L^-, S^-, S'')$, *otherwise,*

where $S' \cup S'' = S^=$ *and* $d(x,y) = \delta$ *for any* $x \in S'$ *and* $y \in S''$. *In particular, if d is Robinsonian, then there exists an order compatible with d that refines the quasi-order* $\mathbf{R}_0$ *computed by the procedure divide(X).*

Note that for ultrametrics the set L is empty, i.e., only case (a) of Theorem 1 occurs.

We now describe in detail how to refine the quasi-order $\mathbf{R}_0$ to obtain a quasi-order compatible with d. The strategy is quite standard. It derives directly from the definition of a compatible order and appears, implicitly at least, in many sources. See, for the first development by Mirkin and Rodin (1984, pp. 63-65), with the superimposing procedure, or Batbedat (1990, p. 69) and (Durand 1989). The quasi-order is constructed recursively (initially, $\mathbf{R} = \mathbf{R}_0$), using the following basic fact:

If $\mathbf{R} = (B_1, \ldots, B_m)$ *is the current quasi-order, and* $\langle$ *is any order refining* $\mathbf{R}$ *and compatible with d, then for any reference point* $b \in B_i$ $(i = 1, \ldots, m)$ *and two objects* $x,y \in B_j$ *such that* $d(b,x) < d(b,y)$, *then* $x \langle y$ *if* $j > i$ *and* $y \langle x$ *if* $i > j$.

Therefore, we must repeatedly refine $\mathbf{R}$ with respect to the quasi-orders R_b, $b \in X$. If the reference point b belongs to the block B_i of a current quasi-order $\mathbf{R} = (B_1, \ldots, B_m)$, then return a new ordered partition whose blocks are B_i and intersections of the blocks $B_1, \ldots, B_{i-1}$, $B_{i+1}, \ldots, B_m$ with the spheres $S_{r_1}(b), \ldots, S_{r_k}(b)$ centered at b (here $r_1 < \ldots < r_k$ are the distinct values taken by the distances from b to the remaining objects). We can get them as well as

the corresponding spheres using the matrix $D_{\leq}$. Employing the idea of *bucket sort* (e.g., Aho, Hopcroft, and Ullman 1974, pp. 77-84) this task can be done in $O(n)$ time. In addition, we get the distances of the objects of B_i to b in the nondecreasing order (thus avoiding the repeated sorting of the dissimilarity matrices of the blocks of occuring quasi-orders). For a given reference point b, denote this procedure by *refine*$(X,\mathbf{R},b)$ and for the resulting quasi-order $(B_{1,1}, \ldots, B_{1,k_1}, \ldots, B_{i-1,1}, \ldots, B_{i-1,k_{i-1}}, B_i, B_{i+1,1}, \ldots, B_{i+1,k_{i+1}}, \ldots, B_{m,1}, \ldots, B_{m,k_m})$ we will reserve the same name $\mathbf{R}$. So, we will simply write in this case $\mathbf{R} :=$ *refine*$(X,\mathbf{R},b)$. In addition, the class B_i is included into a list $\mathbf{L}$, empty at the beginning.

We apply the refining procedure by sweeping the objects from left-to-right in accordance with the current quasi-order $\mathbf{R}$. The complexity of such a sweeping procedure is clearly $O(n^2)$. The blocks introduced in the list $\mathbf{L}$ represent an ordered partition (i.e., a quasi-order $\mathbf{R}^*$ of X). Note that the quasi-order $\mathbf{R}$ obtained after the sweeping of all objects of X refines $\mathbf{R}^*$, because some of the blocks of $\mathbf{L}$ can be further split into subblocks. Therefore, we must sweep separately each block introduced in the list $\mathbf{L}$ and consisting of at least two subblocks. A formal description of this procedure is given below.

procedure refine$(X,\mathbf{R}_0)$

Input: a quasi-order $\mathbf{R}_0 = (B_1, \ldots, B_m)$ and a dissimilarity d of X;
Output: a quasi-order $\mathbf{R}$ of X;
Initialize: $\mathbf{L} = \varnothing$; $\mathbf{R} := \mathbf{R}_0$;

```
begin
    if m = 1 then return R;
    else i = 1;
    until i = m do
        begin
            for any b ∈ B_i do
            begin
                R := refine(X,R,b);L := L ∪ {B_i};
                m := ''the number of blocks of R'';
                i := ''the number of blocks of R left from B_i'' + 1;
            end
        end
    *L := (L_1, ..., L_k)*
    return R := (refine(L_1,R), ..., refine(L_k,R));
end;
```

Lemma 5. *For any two distinct blocks B_i and B_j of the quasi-order $\mathbf{R}$ obtained by the procedure refine$(X,\mathbf{R}_0)$ and any objects $x,y \in B_i$ and $z \in B_j$ the equality $d(x,z) = d(y,z)$ holds.*

Proof. Suppose the contrary. Consider the last recursive call of the procedure *refine* when the objects x, y, and z belong to a common block B_k of the current quasi-order $\mathbf{R}$ (if already in $\mathbf{R}_0$ these objects are in different blocks, then the first application of the procedure *refine*$(X,\mathbf{R},z)$ will assign the objects x and y to different blocks). Again, the procedure *refine*$(B_k,\mathbf{R},z)$ will put x and y in different blocks (necessarily, we have a recursive call of this procedure, because the resulting quasi-order B_k consists of at least two blocks). This conclusion contradicts our initial assumption. ■

We are now ready to describe the complete procedure for recognizing Robinsonian dissimilarities. It consists of the following steps.

procedure robinson(X)

Input: a dissimilarity d on a set X objects given by the matrix D;
Output: an order $\mathbf{R}$ of X compatible with d if d is Robinsonian and the answer "d IS NOT ROBINSONIAN" otherwise;
Initialize: construct the matrix $D_{\leq}$ by arranging the rows of the matrix D in nondecreasing order;

begin
  if $|X| = 1$ **then return** $\mathbf{R} = \{X\}$;
  else do
    begin
      $\mathbf{R}_0$:= *divide*(X);
      $\mathbf{R}$:= *refine*$(X,\mathbf{R}_0)$;
      * **let** $\mathbf{R} := (B_1, \ldots, B_m)$*
      if the quasi-order $\mathbf{R}$ is not compatible with d,
      then return "d IS NOT ROBINSONIAN;"
      else return $\mathbf{R}$:= (*robinson*$(B_1), \ldots,$ *robinson*(B_m));
    end
  end;

Theorem 2. *The procedure robinson*(X) *correctly recognizes in* $O(n^3)$ *time and with* $O(n^2)$ *space whether a dissimilarity d on a set of n objects is Robinsonian.*

Proof. The correctness follows readily from Theorem 1 and the method of refining the quasi-order $\mathbf{R}_0$. Using induction, one can prove that any order of X compatible with d and refining $\mathbf{R}_0$ will refine any quasi-order $\mathbf{R}$ occuring in the procedure *refine*$(X,\mathbf{R}_0)$. Therefore, if as a result we obtain a quasi-order not compatible with d, then d is not Robinsonian. Otherwise, if the resulting quasi-order is compatible with d, then as already noted, d is Robinsonian if and only if its restrictions on blocks of the quasi-order are Robinsonian dissimilarities. This concludes the proof of correctness.

Now we focus on the complexity of the procedure *robinson*(X). The procedure *divide* applied to a set with k objects obviously has the complexity $O(k^2)$ in the worst case. Therefore, the total complexity of this procedure is $O(n^3)$. Now, as already noted, one call of the procedure *refine*(X,**R**,b) takes $O(n)$ time. In particular, one sweep of the set of objects in the procedure *refine*(X,$\mathbf{R}_0$) can be performed in $O(n^2)$ time. Since each time we repeat the same procedure with the blocks of the list **L** only in the case when new blocks appear, during the entire execution of the algorithm, there are at most n such sweeps of the list of objects. Therefore, the refining procedure requires at most $O(n^3)$ operations. Finally, we will check in $O(n^2)$ time whether a given quasi-order is compatible with d. (In fact, because of the properties of the quasi-orders obtained, one can perform this verification only once, when we will get a total order.) This observation shows that the total complexity of the algorithm is $O(n^3)$. ■

4. Examples

Continuing, we apply the algorithm to some data sets. First consider two special examples, when only one or another procedure is used.

Example 1. The dissimilarity d is given below.

0	4	4	5	5	4	4	4	5
4	0	3	5	5	3	2	1	4
4	3	0	5	5	4	3	3	4
5	5	5	0	6	5	5	5	5
5	5	5	6	0	5	5	5	5
4	3	4	5	5	0	3	3	4
4	2	3	5	5	3	0	1	4
4	1	3	5	5	3	1	0	4
5	4	4	5	5	4	4	4	0

In this example after each call of the procedure *divide*, a quasi-order compatible with d is returned (therefore, the procedure *refine* does not create additional blocks). Namely, each of them consists of three blocks (S^-, L^-, C), where S^- and C have only one object each. Here are the quasi-orders obtained after each iteration:

({5}, {6,7,2,9,3,1,8}, {4}),
({5}, {1}, {6,7,2,3,8}, {9}, {4}),
({5}, {1}, {6}, {7,2,8}, {3}, {9}, {4}),
({5}, {1}, {6}, {7}, {8}, {2}, {3}, {9}, {4}).

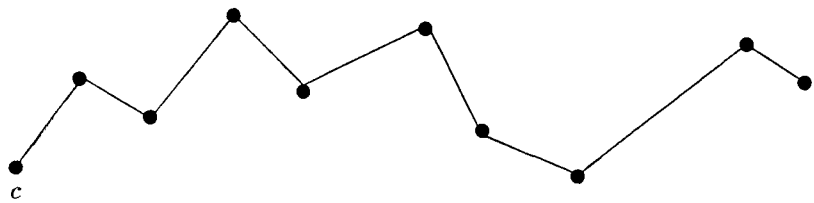

Figure 2. Illustrating Example 2.

The final order (5,1,6,7,8,2,3,9,4) is compatible with d, and thus, d is Robinsonian. In this case, the complexity of the algorithm is $O(n^3)$. In fact, in this example the quasi-orders occuring on each iteration have the most nonbalanced blocks.

Example 2. If all the distances between the objects are distinct, then after the first refining procedure we will get an order. If it is compatible with the dissimilarity, then the positive answer is returned; otherwise, the answer will be "no." The complexity in this case is $O(n^2)$. The same holds true under the weaker assumption that C consists of a single object c and that the distances from c to all other objects are distinct. See Figure 2 for such an example; here the objects are points in the plane, and the Euclidean distance is the dissimilarity measure. In this example the algorithm returns an order which coincides with that generated by a path without self-intersections.

Example 3. Consider the dissimilarity defined by the following matrix:

0	5	5	1	4	5	5	5	5	1
5	0	5	5	3	5	5	1	2	5
5	5	0	5	5	1	1	5	1	5
1	5	5	0	3	5	5	5	5	1
4	3	5	3	0	5	5	3	4	3
5	5	1	5	5	0	1	5	2	5
5	5	1	5	5	1	0	5	2	5
5	1	5	5	1	5	5	0	2	5
5	2	1	5	4	2	2	2	0	5
1	5	5	1	3	5	5	5	5	0

There are several possibilities for selecting the object p. First assume that the algorithm selects $p = 2$. After the first division, the following quasi-order is obtained:

$$(S_1 = \{1,4,10\}, L_1 = \{5\}, C = \{2,8\}, L_2 = \{9\}, S_2 = \{3,6,7\}).$$

Further, applying the refining procedure, we will get the following quasi-order

$$(\{1\},\{4,10\},\{5\},\{2,8\},\{9\},\{3\},\{6,7\}).$$

compatible with d. This result shows that d is Robinsonian (but not strongly-Robinsonian) and has 16 compatible orderings. We present one of them (1,10,4,5,8,2,9,3,7,6) and the corresponding Robinsonian matrix:

0	1	1	4	5	5	5	5	5	5
1	0	1	3	5	5	5	5	5	5
1	1	0	3	5	5	5	5	5	5
4	3	3	0	3	3	4	5	5	5
5	5	5	3	0	1	2	5	5	5
5	5	5	3	1	0	2	5	5	5
5	5	5	4	2	2	0	1	2	2
5	5	5	5	5	5	1	0	1	1
5	5	5	5	5	5	2	1	0	1
5	5	5	5	5	5	2	1	1	0

Otherwise, if object 1 is selected as p, then after the division step we have the quasi-order

$$(S_1 = \{2,3,6,7,8,9\}, L_1 = \{5\}, C = \{1,4,10\}),$$

while after the refining procedure the quasi-order

$$(\{6,7\},\{3\},\{9\},\{2,8\},\{5\},\{4,10\},\{1\})$$

inverse to the first one is returned.

5. Extensions

In this section we present three related recognition problems, which can be solved using our approach. Namely, we deal with the problem of generating all orders compatible with a given Robinsonian dissimilarity, that of finding (if it exists) an order compatible with several dissimilarities defined on a common set of objects, and the problem of recognizing strongly-Robinsonian dissimilarities.

Our test can be easily adapted to check whether a dissimilarity d on X is strongly-Robinsonian. Indeed, it suffices in procedure *robinson*(X) after verifying if d is compatible with the quasi-order $\mathbf{R}$, to perform the same control for the strong compatibility condition. This procedure can be executed within the same time bounds. For this purpose, recall the equivalent definition of an order $\langle$ strongly-compatible with d (see, for example, Critchley and Fichet 1994): if $x \langle y \langle z$ and $d(x,z) = d(y,z)$, then $d(x,t) = d(y,t)$ for any $t \rangle z$, and if $d(x,z) = d(x,y)$, then $d(t,z) = d(t,y)$ for any $t \langle x$. If we already know that the matrix D is Robinsonian, then one can verify these conditions only locally. Indeed, if $i < j < k$ and $d_{ik} = d_{jk}$, then it is enough to verify the condition $d_{ik+1} = d_{jk+1}$. If both conditions are fulfilled, then finally we will get a strongly-Robinsonian dissimilarity, otherwise the answer will be "no." To justify this claim, note that all the results established before remain valid if we replace "Robinsonian" by "strongly-Robinsonian" and "compatible" by "strongly-compatible."

To generate all compatible orders (as for ultrametrics, the number of such orders can be exponential in n), we can apply Theorem 1. According to this result, it suffices to find recursively all compatible orders of the sets $S^=$ and $X - S^= = S^- \cup L^- \cup C \cup L^+ \cup S^+$ and to combine them in the following manner. Pick a compatible order of $S^=$ and split it into two segments $S' = [a,b']$ and $S'' = [b'',c]$, so that $d(b',b'') = \delta$. Then we just add the segments S' and S'' from left and right to any order of $X - S^=$ compatible with d. Concatenating the respective orders, we will get an order of X compatible with d.

Our approach can be modified to solve the following "consensus"-type problem: given k dissimilarities $d_1, \ldots, d_k$ on a set X of n objects, find (if it exists) an order on X compatible with all dissimilarities $d_1, \ldots, d_k$. For this purpose compute the diameters δ_i of X with respect to all d_i, $i = 1, \ldots, k$. For $x \in X$, set $S(x) = \cap_{i=1}^{k} S_{\delta_i}(x)$. At the next step we find an object p with the largest set $S(p)$. Let $S := S(p)$. If S is empty, then there does not exist any common compatible order. Indeed, if such an order exists and u and v are the respective first and last elements, then $d_i(u,v) = \delta_i$, $i = 1, \ldots, k$. In particular, $S(u) \neq \emptyset$ and $S(v) \neq \emptyset$. Now, assume $|S| > 0$. As in the procedure *divide*(X), compute the set $C = \{c \in X : S(c) = S\}$ and the connected components $K_1, \ldots, K_m$ of the threshold graph G. Again, G has $L \cup S$ as the vertex-set, and two objects x and y are adjacent in G if and only if x and y do not belong simultaneously to L and $d_i(x,y) < \delta_i$ for some i. This observation pertains to the changes in the procedure *divide* (the modifications in the procedure *refine* are straightforward). It is easy to see that all the conclusions of Lemmas 1-5 and Theorem 1 remain valid, if we replace "d" by "$d_1, \ldots, d_k$." Thus, we get an $O(n^3 k)$ time and $O(n^2 k)$ space algorithm to detect whether k given dissimilarities on X have a common compatible order.

References

AHO, A. V., HOPCROFT, J. E., and ULLMAN, J. D. (1974), *The Design and Analysis of Computer Algorithms*, Reading, MA: Addison-Wesley.

BATBEDAT, A. (1990), *Les approches pyramidales dans la classification arborée*, Paris: Masson.

BERTRAND, P. (1986), *Etude de la représentation pyramidale*, Thèse de 3ème cycle, Université Paris IX-Dauphine.

BERTRAND, P. (1992), "Propriétés et caractérisations topologiques d'une représentation pyramidale," *Mathématiques, Informatique et Sciences humaines, 117*, 5-28.

BERTRAND, P., and DIDAY, E. (1985), "A Visual Representation of the Compatibility Between an Order and a Dissimilarity Index: The Pyramids," *Computational Statistics Quarterly, 2*, 31-44.

BOOTH, K. S., and LUEKER, G. E. (1976), "Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms," *Journal of Computational Systems and Sciences, 13*, 335-379.

BROSSIER, G. (1980), "Représentation ordonnée des classification hiérarchiques," *Statistique et Analyse des Données, 5*, 31-44.

CRITCHLEY, F., and FICHET, B. (1994), "The Partial Order by Inclusion of the Principal Classes of Dissimilarity on a Finite Set, and Some of their Basic Properties," in *Classification and Dissimilarity Analysis*, Ed., B. Van Cutsem, Lecture Notes in Statistics, New York: Springer-Verlag, 5-65.

DIDAY, E. (1983), "Croisements, ordres et ultramétriques," *Mathématiques et Sciences humaines, 83*, 31-54.

DIDAY, E.(1986), "Orders and Overlapping Clusters by Pyramids", in *Multidimensional Data Analysis*, Eds., J. de Leeuw, W. Heiser, J. Meulman, and F. Critchley, Leiden: DSWO, 201-234.

DURAND, C. (1989), *Ordres et graphes pseudo-hiérarchiques: théorie et optimisation algorithmique*, Thèse de l'Université de Provence, Marseille.

DURAND, C., and FICHET, B. (1988) "One-to-one Correspondences in Pyramidal Representation: A Unified Approach", in *Classification and Related Methods of Data Analysis*, Ed., H.H. Bock, Amsterdam: North-Holland, 85-90.

FICHET, B. (1986), "Data Analysis: Geometric and Algebraic Structures," in *First World Congress of Bernoulli Society Proceedings, vol.2, Tashkent, USSR, 1986*, Eds., Yu.A. Prohorov and V.V. Sazonov, Utrecht: VNU Science Press, 123-132.

FULKERSON, D. R. and GROSS, O. A. (1965), "Incidence Matrices and Interval Graphs," *Pacific Journal of Mathematics, 15*, 835-855.

GOLUMBIC, M. C. (1980), *Algorithmic Graph Theory and Perfect Graphs*, New York: Academic Press.

HUBERT, L. J. (1974), "Some Applications of Graph Theory and Related Nonmetric Techniques to Problems of Approximate Seriation: The Case of Symmetric Proximity Measures," *British Journal of Mathematical Statistics and Psychology, 27*, 133-153.

KENDALL, D. G. (1969), "Incidence Matrices, Interval Graphs and Seriation in Archaeology," *Pacific Journal of Mathematics, 28*, 565-570.

MIRKIN, B. (1996), *Mathematical Classification and Clustering*, Dordrecht: Kluwer.

MIRKIN, B., and RODIN, S. (1984), *Graphs and Genes*, Berlin: Springer-Verlag.

ROBINSON, W. S. (1951), "A Method for Chronologically Ordering Archaeological Deposits", *American Antiquity, 16*, 293-301.