MODULES IN ROBINSON SPACES *

MIKHAEL CARMONA[†‡], VICTOR CHEPOI[†], GUYSLAIN NAVES[†], AND PASCAL PRÉA[†‡]

Abstract. A *Robinson space* is a dissimilarity space (X, d) (i.e., a set X of size n and a dissimilarity d on X) for which there exists a total order $<$ on X such that $x < y < z$ implies that $d(x, z) \geq \max\{d(x, y), d(y, z)\}$. Recognizing if a dissimilarity space is Robinson has numerous applications in seriation and classification. An *mmodule* of (X, d) (generalizing the notion of a module in graph theory) is a subset M of X which is not distinguishable from the outside of M, i.e., the distance from any point of $X \setminus M$ to all points of M is the same. If p is any point of X, then $\{p\}$ and the maximal by inclusion mmodules of (X, d) not containing p define a partition of X, called the *copoint partition*. In this paper, we investigate the structure of mmodules in Robinson spaces and use it and the copoint partition to design a simple and practical divide-and-conquer algorithm for recognition of Robinson spaces in optimal $O(n^2)$ time.

Key words. Robinson dissimilarity, Seriation, Classification, Mmodule, Divide-and-conquer.

AMS subject classifications. 68R01, 05C85, 68P10

1. Introduction.

A major issue in classification and data analysis is to visualize simple geometrical and relational structures between objects based on their pairwise distances. Many applied algorithmic problems ranging from archeological dating through DNA sequencing and numerical ecology to sparse matrix reordering and overlapping clustering involve ordering a set of objects so that closely coupled elements are placed near each other. The classical *seriation problem*, introduced by Robinson [32] as a tool to seriate archeological deposits, asks to find a simultaneous ordering (or permutation) of the rows and the columns of the dissimilarity matrix so that small values should be concentrated around the main diagonal as closely as possible, whereas large values should fall as far from it as possible. This goal is best achieved by considering the so-called Robinson property: a dissimilarity matrix has the *Robinson property* if its values increase monotonically in the rows and the columns when moving away from the main diagonal in both directions. A *Robinson matrix* is a dissimilarity matrix which can be transformed by permuting its rows and columns to a matrix having the Robinson property. The permutation which leads to a matrix with the Robinson property is called a *compatible order*. Computing a compatible order can be viewed as the two-dimensional version of the sorting problem. In this paper, we present a simple and practical divide-and-conquer algorithm for computing a compatible order and thus recognizing Robinson matrices in optimal $O(n^2)$ time.

1.1. Related work.

Due to the importance in seriation and classification, the algorithmic problem of recognizing Robinson dissimilarities/matrices attracted the interest of many authors and several polynomial time recognition algorithms have been proposed. The existing algorithms can be classified into *combinatorial* and *spectral*. All combinatorial algorithms are based on the correspondence between Robinson dissimilarities and interval hypergraphs. The main difficulty arising in recognition algorithms is the existence of several compatible orders. Historically, the first recognition algorithm was given in 1984 by Mirkin and Rodin [28] and consists in testing if the hypergraph of balls is an interval hypergraph; it runs in $O(n^4)$ time. Chepoi and Fichet [8] gave a simple divide-and-conquer algorithm running in $O(n^3)$ time. The algorithm divides the set of points into subsets and refines the obtained subsets into classes to which the recursion can be applied. Seston [34] presented another $O(n^3)$ time algorithm, by using threshold graphs. In [33], he improved the complexity of his algorithm to $O(n^2 \log n)$ by using PQ-trees. Finally, in 2014 Préa and Fortin [30] presented an algorithm running in optimal $O(n^2)$ time. The efficiency of the algorithm of [30] is due to the use of the PQ-trees of Booth and Lueker [5] as a data structure for encoding all compatible orders. Even if optimal, the algorithm of [30] is far from being simple. Subsequently, two new recognition algorithms were proposed by Laurent and Seminaroti: in [25] they presented an algorithm of complexity $O(\alpha \cdot n)$ based on classical LexBFS traversal and divide-and-conquer (where α is the depth of the recursion tree, which is at most the number of distinct elements of the input matrix), and in [26] they presented an $O(n^2 \log n)$ algorithm, which extends

* This research was supported in part by ANR project DISTANCIA (ANR-17-CE40-0015).
The project leading to this publication has received funding from Excellence Initiative of Aix-Marseille - A*MIDEX (Archimedes Institute AMX-19-IET-009), a French "Investissements d'Avenir" Programme

[†]LIS, Aix-Marseille Université, CNRS, and Université de Toulon, Marseille, France ({mikhael.carmona, victor.chepoi, guyslain.naves, pascal.prea}@lis-lab.fr)

[‡]École Centrale Marseille, Marseille, France

LexBFS to weighted matrices and is used as a multisweep traversal. Laurent, Seminaroti and Tanigawa [27] presented a characterization of Robinson matrices in terms of forbidden substructures, extending the notion of asteroidal triples in graphs to weighted graphs. More recently, Aracena and Thraves Caro [1] presented a parametrized algorithm for the NP-complete problem of recognition of Robinson incomplete matrices. Armstrong et al. [2] presented an optimal $O(n^2)$ time algorithm for the recognition of strict circular Robinson dissimilarities (Hubert et al. [21] defined circular seriation first and it was studied also in the papers [31] and [23]).

The spectral approach was introduced by Atkins et al. [3] and was subsequently used in numerous papers (see, for example, [17] and the references therein). The method is based on the computation of the second smallest eigenvalue and its eigenvector of the Laplacian of a similarity matrix A, called the *Fiedler value* and the *Fiedler vector* of A. Atkins et al. [3, Theorems 3.2 & 3.3] proved that if A is Robinson, then it has a monotone Fiedler vector and if A is Robinson with a Fiedler value and a Fiedler vector with no repeated values, then the two permutations of the Fiedler vector in which the coordinates are strictly increasing (respectively, decreasing) are the only two compatible orders of A. For similarity matrices for which the Fiedler vector has repeated values, Atkins et al. [3] recursively apply the algorithm to each submatrix of A defined by coordinates of the Fiedler vector with the same value. In this case, they also use PQ-trees to represent the compatible orders. This leads to an algorithm of complexity $O(nT(n) + n^2 \log n)$ to recognize if a similarity matrix is Robinson, where $T(n)$ is the complexity of computing the Fiedler vector of a matrix. The Fiedler vector is computed by the Lanczos algorithm, which is an iterative numerical algorithm that at each iteration performs a multiplication of the input matrix A by a vector.

Real data contain errors, therefore the dissimilarity matrix can be measured only approximatively and fails to satisfy the Robinson property. In this case, we are lead to the problem of approximating a dissimilarity D by a Robinson dissimilarity R. As an error measure one can use the usual ℓ_p-distance $||D - R||_p$ between two $n \times n$ matrices. This ℓ_p-fitting problem has been shown to be NP-hard for $p = 1$ [4] and for $p = \infty$ in [9]. Various heuristics for this optimization problem have been considered in [6, 20, 22] and papers cited therein. The approximability of this fitting problem for any $1 \le p < \infty$ is open. Chepoi and Seston [10] presented a factor 16 approximation for the ℓ_∞-fitting problem. For a similarity matrix A, Ghandehari and Janssen [18] introduced a parameter $\Gamma_1(A)$ and showed that one can construct a Robinson similarity R (with the same order of lines and columns as A) such that $||A - R||_1 \le 26\Gamma_1(A)^{\frac{1}{3}}$. The result of Atkins et al. [3] in the case of the Fiedler vectors with no repeated values was generalized by Fogel et al. [17] to the case when the entries of A are subject to a uniform noise or some entries are not given. Basic examples of Robinson dissimilarities are the ultrametrics. Similarly to the classical bijection between ultrametrics and hierarchies, there is a one-to-one correspondence between Robinson dissimilarities and pseudo-hierarchies due to Diday [12] and Durand and Fichet [13]. Pseudo-hierarchies are now classical examples of classification with overlapping classes.

1.2. Paper's organization. The paper is organized as follows. The main notions related to the Robinson property are given in Section 2. In Section 3, we introduce mmodules and copoints of a dissimilarity space (X, d) and present their basic properties. In particular, we show that all copoints of a given point p define a partition of $X \setminus \{p\}$. In Section 4, we investigate the copoint partitions and the compatible orders for flat and conical Robinson spaces. In Section 5, we investigate the properties of copoint partitions and their (extended) quotients in general Robinson spaces. In Section 6, we introduce the concept of proximity pre-order for an (unknown) compatible order. We show that for extended quotients this pre-order is an order and we show how to retrieve a compatible order from this proximity order. The results of these sections are used in the divide-and-conquer algorithm, described and analyzed in Section 7.

2. Preliminaries.

2.1. Robinson dissimilarities. Let $X = \{p_1, \dots, p_n\}$ be a set of n elements, called *points*. A *dissimilarity* on X is a symmetric function d from X^2 to the nonnegative real numbers such that $d(x, y) = d(y, x) \ge 0$ and $d(x, y) = 0$ if $x = y$. Then $d(x, y)$ is called the *distance* between x, y and (X, d) is called a *dissimilarity space*. A partial order on X is called *total* if any two elements of X are comparable. Since we will mainly deal with total orders, we abbreviately call them *orders*. A dissimilarity d and an order $<$ on X are called *compatible* if $x < y < z$ implies that $d(x, z) \ge \max\{d(x, y), d(y, z)\}$. If d and $<$ are compatible, then d is also compatible with the order $<^{\text{op}}$ opposite to $<$. If a dissimilarity space (X, d) admits a compatible order, then

d is said to be *Robinson* and (X, d) is called a *Robinson space*. Equivalently, (X, d) is Robinson if its distance matrix $D = (d(p_i, p_j))$ can be symmetrically permuted so that its elements do not decrease when moving away from the main diagonal along any row or column. Such a dissimilarity matrix D is said to have the *Robinson property* [11, 12, 13, 20]. If $Y \subset X$, we denote by $(Y, d|_Y)$ (or simply by (Y, d)) the dissimilarity space obtained by restricting d to Y; we call (Y, d) a *subspace* of (X, d). If (X, d) is a Robinson space, then any subspace (Y, d) of (X, d) is also Robinson and the restriction of any compatible order $<$ of X to Y is compatible. Given two dissimilarity spaces (X', d') and (X, d), a map $\varphi : X' \to X$ is an *isometric embedding* of (X', d') in (X, d) if for any $x, y \in X'$ we have $d(\varphi(x), \varphi(y)) = d'(x, y)$, i.e., if (X', d') can be viewed as a subspace of (X, d).

The *ball* of radius $r \geq 0$ centered at $x \in X$ is the set $B_r(x) := \{y \in X : d(x, y) \leq r\}$. The *diameter* of a subset Y of X is $\mathrm{diam}(Y) := \max\{d(x, y) : x, y \in Y\}$ and a pair $x, y \in Y$ such that $d(x, y) = \mathrm{diam}(Y)$ is called a *diametral pair* of Y. A point x of Y is called *non-diametral* if x does not belong to a diametral pair of Y. From the definition of a Robinson dissimilarity follows that d is Robinson if and only if there exists an order $<$ on X such that all balls $B_r(x)$ of (X, d) are intervals of $<$. Moreover, this property holds for all compatible orders. Basic examples of Robinson dissimilarities are the ultrametrics, thoroughly used in phylogeny. Recall, that d is an *ultrametric* if $d(x, y) \leq \max\{d(x, z), d(y, z)\}$ for all $x, y, z \in X$. Another example of a Robinson space is provided by the standard *line-distance* between n points $p_1 < \ldots < p_n$ of $\mathbb{R}$. Any line-distance has exactly two compatible orders: the order $p_1 < \ldots < p_n$ defined by the coordinates of the points and its opposite. If a Robinson space (X, d) has only two compatible orders $<$ and $<^{op}$, then (X, d) is said to be *flat*. Line-distances are flat but the converse is not true. A dissimilarity space (X, d) is *conical* if there exists $\delta > 0$ and $p \in X$ such that $d(p, x) = \delta$ for any $x \in X \setminus \{p\}$. Since p has the same distance δ to all points of $X \setminus \{p\}$, (X, d) is a *cone* over $(X \setminus \{p\}, d)$ with *apex* p.

2.2. Partitions and pre-orders. Let (X, d) be a dissimilarity space. A *partition* of X is a family of sets $\mathcal{P} = \{B_1, \ldots, B_m\}$ such that $B_i \cap B_j = \varnothing$ for any $i \neq j$ and $\bigcup_{i=1}^k B_i = X$. The sets $B_1, \ldots, B_m$ are called the *classes* of $\mathcal{P}$. A *pre-order* is a partial order $\prec$ on X for which incomparability is transitive. A partial order $\prec$ on X is a pre-order exactly when there exists an ordered partition $\mathcal{R} = (B_1, \ldots, B_m)$ of X such that for $x \in B_i$ and $y \in B_j$ we have $x \prec y$ if and only if $i < j$. Consequently, we will also view a pre-order $\prec$ as an ordered partition $\mathcal{R} = (B_1, \ldots, B_m)$. A partial order $\prec'$ *extends* a partial order $\prec$ on X if $x \prec y$ implies $x \prec' y$ for all $x, y \in S$.

A partition $\mathcal{P} = \{B_1, \ldots, B_m\}$ of X is called a *stable partition* if for any $i \neq j$ and for any three points $x, y \in B_i$ and $z \in B_j$, we have $d(z, x) = d(z, y)$. If a partition $\mathcal{P}$ is not stable, then we can transform it into a stable partition by applying the operation of *partition refinement*, which proceeds as follows. The algorithm maintains the current partition $\mathcal{P}$ and for each class B of $\mathcal{P}$ maintains the set $Z(B)$ of all points outside B which still have to be processed to refine B. While $\mathcal{P}$ contains a class B with nonempty $Z(B)$, the algorithm pick any point z of $Z(B)$ and partition B into maximal classes that are not distinguishable from z: i.e., for any such new class B' and any $x, x' \in B'$ we have $d(x, z) = d(x', z)$. Finally, the algorithm removes B from $\mathcal{P}$ and insert each new class B' in $\mathcal{P}$ and sets $Z(B') := (B \setminus B') \cup (Z(B) \setminus \{z\})$. Notice that each class B is partitioned into subclasses by comparing the distances of points of B to the point $z \notin B$ and such distances never occur in later comparisons. Also, if the final stable partition has classes $B'_1, \ldots, B'_t$, then the distances between points in the same class B'_i are never compared to other distances. Later we will show how to implement this algorithm with total complexity $O(|X|^2 - \sum_{i=1}^t |B'_i|^2)$.

3. Mmodules in dissimilarity spaces. In this section, we introduce and investigate the notion of mmodule. As one can see from their use in this paper, our motivation for introducing them stems from the property of classes in stable partitions: the points of the same class C cannot be distinguished from the outside, i.e., for any two points $x, y \in C$ and any point $z \notin C$, the equality $d(z, x) = d(z, y)$ holds. After having obtained the main properties of mmodules in general dissimilarities presented in Subsection 3.1, we discovered that our mmodules coincide with "clans" in symmetric 2-structures, defined and investigated by Ehrenfeucht and Rozenberg [15, 16]. Since their theory is developed in a more general non-symmetric setting, we prefer to give a self-contained presentation of elementary properties of mmodules. Applying an argument from abstract convexity, we deduce that for each point p all maximal by inclusion mmodules not containing p together with p define a partition of X. This partition is used in our divide-and-conquer algorithm for recognizing Robinson spaces.

3.1. Mmodules. Let (X, d) be a dissimilarity space. A set $M \subseteq X$ is called an *mmodule* (a *metric module* or a *matrix module*, pronounced [ɛm 'mɒdjuːl]) if for any $z \in X \setminus M$ and all $x, y \in M$ we have $d(z, x) = d(z, y)$. In graph theory, the subgraphs indistinguishable from the outside are called modules (see [14, 19]), explaining our choice of the term "mmodule". Denote by $\mathcal{M} = \mathcal{M}(X, d)$ the set of all mmodules of (X, d). Trivially, $\varnothing, X$, and $\{p\}, p \in X$ are mmodules; we call them *trivial mmodules*. An mmodule M is called *maximal* if M is a maximal by inclusion mmodule different from X. We continue with the basic properties of mmodules.

PROPOSITION 3.1. *The set $\mathcal{M} = \mathcal{M}(X, d)$ has the following properties:*

(i) $M_1, M_2 \in \mathcal{M}$ implies that $M_1 \cap M_2 \in \mathcal{M}$;
(ii) if $M \in \mathcal{M}$ and $M' \subset M$, then $M' \in \mathcal{M}$ if and only if M' is an mmodule of (M, d);
(iii) if $M_1, M_2 \in \mathcal{M}$ and $M_1 \cap M_2 \neq \varnothing$, then $M_1 \cup M_2, M_1 \setminus M_2, M_2 \setminus M_1, M_1 \triangle M_2 \in \mathcal{M}$;
(iv) the union $M_1 \cup M_2$ of two intersecting maximal mmodules $M_1, M_2 \in \mathcal{M}$ is X;
(v) if M_1 and M_2 are two disjoint maximal mmodules and M is a nontrivial mmodule contained in $M_1 \cup M_2$, then either $M \subset M_1$ or $M \subset M_2$;
(vi) if $M_1, M_2 \in \mathcal{M}$ and $M_1 \cap M_2 = \varnothing$, then $d(u, v) = d(u', v')$ for any (not necessarily distinct) points $u, u' \in M_1$ and $v, v' \in M_2$;
(vii) if $\mathcal{M}'$ is any partition of X into mmodules, then $\mathcal{M}'$ is a stable partition.

Proof. To (i): Pick any $x \notin M_1 \cap M_2$ and $u, v \in M_1 \cap M_2$. If $x \notin M_1 \cup M_2$, then $d(x, u) = d(x, v)$ since $M_1, M_2 \in \mathcal{M}$. If $x \in M_1 \cup M_2$, say $x \in M_2 \setminus M_1$, then $d(x, u) = d(x, v)$ since $M_1 \in \mathcal{M}$.

To (ii): First, let M' be an mmodule of (X, d). This implies that $d(x, u) = d(x, v)$ for any $x \in M \setminus M'$ and $u, v \in M'$, thus M' is an mmodule of (M, d). Conversely, let M' be an mmodule of (M, d) and we assert that M' is an mmodule of (X, d). Pick any $x \in X \setminus M'$ and $u, v \in M'$. If $x \in X \setminus M$, then $d(x, u) = d(x, v)$ since $u, v \in M' \subset M$ and M is an mmodule. If $x \in M \setminus M'$, then $d(x, u) = d(x, v)$ since M' is an mmodule of (M, d) and we are done.

To (iii): If $M_1 \cup M_2 = X$, we are done. Otherwise, pick any $x \in X \setminus (M_1 \cup M_2)$ and $u, v \in M_1 \cup M_2$. If $u, v \in M_1$ or $u, v \in M_2$, then $d(x, u) = d(x, v)$ because $M_1, M_2 \in \mathcal{M}$. Thus, let $u \in M_1 \setminus M_2$ and $v \in M_2 \setminus M_1$. Pick any $w \in M_1 \cap M_2$. Then $d(x, u) = d(x, w)$ and $d(x, v) = d(x, w)$ since M_1 and M_2 are mmodules. Consequently, $d(x, u) = d(x, v)$ and thus $M_1 \cup M_2 \in \mathcal{M}$.

Since $M_1, M_2 \in \mathcal{M}$, for any $u, v \in M_1 \setminus M_2, u', v' \in M_2 \setminus M_1, y \in M_1 \cap M_2$, and $x \in X \setminus (M_1 \cup M_2)$, we have $d(x, u) = d(x, v) = d(x, y) = d(x, u') = d(x, v')$ and $d(u, u') = d(v, v') = d(u, y) = d(v, y) = d(u', y) = d(v', y)$. This shows that $M_1 \setminus M_2, M_2 \setminus M_1$, and $M_1 \Delta M_2$ are mmodules.

To (iv): This is a direct consequence of (iii) and the definition of maximal mmodules.

To (v): Suppose that there exist $y \in M \cap M_1$ and $z \in M \cap M_2$. We assert that $M' := M_1 \cup (M \cap M_2) = M_1 \cup M$ is a nontrivial mmodule, contradicting the maximality of M_1. Indeed, pick any $x \in X \setminus M'$ and $u, v \in M'$. If $u, v \in M_1$ or $u, v \in M \cap M_2$, then $d(x, u) = d(x, v)$ because $M_1, M \in \mathcal{M}$ and $x \notin M_1 \cup M$. So, let $u \in M_1$ and $v \in M \cap M_2$. Since M_1 is an mmodule and $u, y \in M_1$, we conclude that $d(x, u) = d(x, y)$. Analogously, since $M \in \mathcal{M}$ and $v, y, z \in M$ and $x \notin M$, $d(x, v) = d(x, z) = d(x, y)$. Consequently, we get $d(x, u) = d(x, v)$ also in this case. This shows that $M_1 \cup M \in \mathcal{M}$. If $M_1 \cup M = X$, this would imply that $M_2 \subset M$, contradicting the fact that M_2 is a maximal mmodule. Consequently, M is contained either in M_1 or in M_2.

To (vi): Since M_2 is an mmodule and $u \notin M_2$, $d(u, v) = d(u, v')$. Since M_1 is an mmodule and $v' \notin M_1$, $d(v', u) = d(v', u')$. Consequently, $d(u, v) = d(u', v')$.

To (vii): This follows from the definition of mmodules and the fact that $\mathcal{M}'$ partitions X. □

By Proposition 3.1(i), $\mathcal{M}$ is closed by intersections, thus $(X, \mathcal{M})$ is a *convexity structure* [35]. Thus for each subset A of X we can define the *convex hull* $\mathrm{mconv}(A)$ of A as the smallest mmodule containing A: $\mathrm{mconv}(A)$ is the intersection of all mmodules containing A. For points $u, v \in X$, we call $\langle u, v\rangle := \{x \in X : d(x, u) \neq d(x, v)\}$ the *interval* between u and v.

LEMMA 3.2. $\langle u, v\rangle \subseteq \mathrm{mconv}(u, v)$.

Proof. Pick x in $\langle u, v\rangle$. If $x \notin \mathrm{mconv}(u, v)$, then $d(x, u') = d(x, v')$ for any $u', v' \in \mathrm{mconv}(u, v)$. This is impossible since $u, v \in \mathrm{mconv}(u, v)$ and $d(x, u) \neq d(x, v)$ by the definition of $\langle u, v\rangle$. □

The converse inclusion is not true. However, the following lemma shows that $\mathcal{M}$ is an *interval convexity* [35] in the following sense:

LEMMA 3.3. *$A \subseteq X$ is an mmodule if and only if $\langle u, v\rangle \subseteq A$ for any two points $u, v \in A$.*

Proof. By Lemma 3.2, $\langle u, v\rangle \subseteq \mathrm{mconv}(u, v)$. Since mconv is a convexity operator, $\mathrm{mconv}(u, v) \subseteq \mathrm{mconv}(A)$. Thus, if $A \in \mathcal{M}$, then $\langle u, v\rangle \subseteq \mathrm{mconv}(A) = A$. Conversely, suppose $\langle u, v\rangle \subseteq A$ for any $u, v \in A$. If A is not an mmodule, there exist $x \in S \setminus A$ and $u, v \in A$ such that $d(x, u) \neq d(x, v)$. But this implies that x belongs to $\langle u, v\rangle \subseteq A$, contrary to the choice of x. □

3.2. Copoint partition. Following the terminology from abstract convexity [24, 35], a *copoint at a point p* (or a *p-copoint*) is any maximal by inclusion mmodule C not containing p; the point p is the *attaching point* of C. The copoints of $\mathcal{M}$ minimally generate $\mathcal{M}$, in the sense that each mmodule M is the intersection of the copoints containing M [35]. Denote by $\mathcal{C}_p$ the set of all copoints at p plus the trivial mmodule $\{p\}$.

LEMMA 3.4. *For any $p \in X$, $\mathcal{C}_p$ defines a partition of X.*

Proof. Pick any copoints C, C' at p. If $C \cap C' \neq \varnothing$, by Proposition 3.1(iii), the union $C \cup C'$ is an mmodule not containing p, contrary to the assumption that C, C' are copoints at p. Since any point $q \neq p$ is an mmodule, q is contained in a copoint at p. Thus $\mathcal{C}_p$ defines a partition of X. □

We call $\mathcal{C}_p := \{C_0 := \{p\}, C_1, \dots, C_k\}$ a *copoint partition* of (X, d) with attaching point p. Then $\mathcal{C}_p$ is called *trivial* if $\mathcal{C}_p$ consists only of the points of X, i.e., $\mathcal{C}_p = \{\{x\} : x \in X\}$, and *cotrivial* if $\mathcal{C}_p = \{\{p\}, X\setminus\{p\}\}$, i.e., all points of $X \setminus \{p\}$ have the same distance to p. Therefore, the copoint partition $\mathcal{C}_p$ is cotrivial if and only if (X, d) is conical with apex p. If $\mathcal{C}_p$ is trivial (equivalently if all non-trivial mmodules of X contain p), then (X, d) is called *p-trivial*. The following result follows directly from the definitions:

LEMMA 3.5. *For a dissimilarity space (X, d), the following holds:*
(i) (X, d) is p-trivial for all $p \in X$ if and only if all mmodules of (X, d) are trivial;
(ii) (X, d) is conical for all $p \in X$ if and only if $d(x, y) = \delta$ for all $x \neq y$ and some $\delta > 0$;
(iii) if (X, d) is conical with apex p, then each mmodule of $(X \setminus \{p\}, d)$ is an mmodule of (X, d).

From Proposition 3.1(vii) it follows that $\mathcal{C}_p$ is a stable partition of X. The partition $\mathcal{C}_p$ can be constructed by applying the partition refinement to the initial partition $\{\{p\}, X \setminus \{p\}\}$.

Let $\mathcal{C}_p = \{C_0 = \{p\}, C_1, \dots, C_k\}$. The *quotient space* $(\mathcal{C}_p, \widehat{d})$ of (X, d) has the classes of $\mathcal{C}_p$ as points and for $C_i, C_j, i \neq j$ of $\mathcal{C}_p$ we set $\widehat{d}(C_i, C_j) := d(u, v)$ for an arbitrary pair $u \in C_i, v \in C_j$.

REMARK 1. *The definition of the quotient space implies that we can partition the rows and the columns of the distance matrix D of (X, d) into sets corresponding to the copoints of $\mathcal{C}_p$, so that after permuting the rows and the columns of D so that we start with the rows and columns corresponding to the first copoint $C_0 = \{p\}$ of $\mathcal{M}$, then to the second copoint C_1 of $\mathcal{C}_p$, etc., the resulting permuted matrix D' has the following nice property: for each pair $C_i, C_j, i \neq j$, of $\mathcal{C}_p$, the entries of D' corresponding to rows from C_i and columns from C_j and rows from C_j and columns from C_i are all equal to $\widehat{d}(C_i, C_j)$. This provides a block decomposition of D' such that in each rectangle not intersecting the main diagonal of D' all entries are equal. The rectangles intersecting the main diagonal are squares defined by the entries located at the intersections of rows and columns corresponding to a copoint C_i. Therefore the recursive call to $C_i \in \mathcal{C}_p$ corresponds to dealing with the dissimilarity space (C_i, d) defined by the entries in this diagonal square. The dissimilarity matrix of $(\mathcal{C}_p, \widehat{d})$ is obtained from D' by replacing each $|C_i| \times |C_j|$ rectangle by a single entry $\widehat{d}(C_i, C_j)$ and contracting each copoint of $\mathcal{C}_p$ to a single point.*

LEMMA 3.6. *The quotient space $(\mathcal{C}_p, \widehat{d})$ is $\{p\}$-trivial.*

Proof. Let $\mathcal{C}_p = \{C_0 = \{p\}, C_1, \dots, C_k\}$ and suppose that $(\mathcal{C}_p, \widehat{d})$ has a non-trivial mmodule M not containing $\{p\}$. For any $C_j, C_{j'} \in M$ and $C_i \in \mathcal{C}_p \setminus M$, we have $\widehat{d}(C_i, C_j) = \widehat{d}(C_i, C_{j'})$. So, if we set $Y := \bigcup_M C_i$, then for any $x, y \in Y$ and $z \in X \setminus Y$, we have $d(z, x) = d(z, y)$. Consequently, Y is an mmodule of (X, d) not containing p, contradicting the maximality of the C_i's. □

3.3. Tree representations of mmodules. A family of subsets $\{M_1, \dots, M_k\}$ of X is a *copartition* of X if $\{X \setminus M_1, \dots, X \setminus M_k\}$ is a partition of X. For a set $M \subseteq X$, let $\overline{M} := X \setminus M$. We denote by $\widehat{\mathcal{M}} := \widehat{\mathcal{M}}(X, d)$ the set of all maximal mmodules of (X, d).

LEMMA 3.7. *$\widehat{\mathcal{M}}$ is a partition or a copartition of X.*

Proof. If the maximal mmodules are pairwise disjoint, then $\widehat{\mathcal{M}}$ is a partition. We assume now that there exist intersecting maximal mmodules M and M'. Then $M \cup M' = X$ by Proposition 3.1(iv). We assert that every pair of maximal mmodules intersects. Let $M_1, M_2 \in \widehat{\mathcal{M}}$ and suppose M_1 and M_2 are disjoint. We may assume $M \cap M_1 \neq \varnothing$. By Proposition 3.1(iv), $M \cup M_1 = X$, and then $M_2 \subseteq M$. By maximality $M_2 = M$, contradicting the fact that M_1 and M_2 are disjoint. Hence any two maximal mmodules M_1 and M_2 intersect, yielding $M_1 \cup M_2 = X$ and $\overline{M_1} \cap \overline{M_2} = \varnothing$.

Let $A = \bigcap \widehat{\mathcal{M}}$ be the intersection of all maximal mmodules. We claim that $\overline{A}$ is an mmodule. If $\overline{A} \neq X$, there exists $b_1, b_2 \in \overline{A}$, $a \in A$, an mmodule M_1 such that $b_1, a \in M_1$ and $b_2 \notin M_1$ and an mmodule M_2 such that $b_2, a \in M_2$ and $b_1 \notin M_2$. So, for $i \in \{1, 2\}$, we have $d(b_i, a) = d(b_i, b_{3-i})$, hence $d(b_1, a) = d(b_2, a)$, proving that $\overline{A}$ is an mmodule. If $\overline{A} \neq X$, it is contained in a maximal mmodule, which contains A by definition, hence $\overline{A} \cup A = X$ is contained in a maximal mmodule, contradiction. Thus, $A = \varnothing$. This proves that $\bigcup\{\overline{M} : M \in \widehat{\mathcal{M}}\} = X$, and we already know that all these sets are disjoint, so $\widehat{\mathcal{M}}$ is a copartition of X. □

LEMMA 3.8. *If $\widehat{\mathcal{M}} = \{M_1, \ldots, M_k\}$ is a copartition, then for any mmodule $M \in \mathcal{M}$, either there is $J \subseteq \{1, 2, \ldots, k\}$ such that $M = \bigcup_{j \in J} \overline{M_j}$ or there is $i \in \{1, 2, \ldots, k\}$ such that $M \subseteq \overline{M_i}$.*

Proof. Let M be an mmodule and suppose that M intersects the complements of two maximal mmodules, say $M \cap \overline{M_1} \neq \varnothing$ and $M \cap \overline{M_2} \neq \varnothing$. Since $M_1 \cap M \neq \varnothing$, by Proposition 3.1(iii), $M_1 \cup M$ is an mmodule which strictly contains M_1. By maximality of M_1, $M_1 \cup M = X$ and $\overline{M_1} \subseteq M$. Consequently, $M = \bigcup\{\overline{M_i} : M \cap \overline{M_i} \neq \varnothing\}$, proving the assertion. □

PROPOSITION 3.9. *Let (X, d) be a dissimilarity space. There is a tree with leaves X and inner nodes labelled by $\cup$ or $\cap$, such that any proper mmodule appears exactly once as in (i) and (ii):*

(i) if a node N is labelled $\cup$, then the set of leaves of any child of N is an mmodule,

(ii) if a node N is labelled $\cap$, then the set of leaves of the union of any proper subset of children of N is an mmodule.

Proof. If $\widehat{\mathcal{M}}$ is a partition, then we define the root to have label $\cup$ and its children are the trees defined inductively for each maximal mmodule. If $\widehat{\mathcal{M}}$ is a copartition, then the root has label $\cap$ and its children are the trees defined inductively for complements of maximal mmodules. By Lemma 3.7, this procedure defines a tree, whence it only remains to prove properties (i) and (ii). These properties hold for maximal mmodules. Pick now a non-maximal mmodule M. By Proposition 3.1(v), if $\widehat{\mathcal{M}}$ is a partition, M is contained in a maximal mmodule M' associated with some child of the root. By induction hypothesis, M is represented in that child. If $\widehat{\mathcal{M}}$ is a copartition, by Lemma 3.8, either M is the union of the complements of maximal mmodules, which corresponds to (ii), or M is strictly contained in the complement of some maximal mmodule M'', and $\overline{M''}$ is represented as a child of the root. By induction hypothesis, M is represented in that child. □

4. Flat and conical Robinson spaces. In this section, we study the copoint partitions in flat Robinson spaces. For conical Robinson spaces we show how to derive compatible orders from compatible orders of subspaces.

4.1. Copoint partitions in flat Robinson spaces. We prove the following property:

PROPOSITION 4.1. *If (X, d) is a flat Robinson space, then either all copoint partitions of (X, d) are trivial or there exists a (unique) non-diametral point p of X such that (X, d) is conical with apex p and all mmodules of $(X \setminus \{p\}, d)$ are trivial.*

Proof. Let $n = |X| > 2$. We order X by a compatible order $q_1 < q_2 < \ldots < q_n$. Let M be an mmodule of (X, d). Let $i = \min\{k \in \{1, \ldots, n\} : q_k \in M\}$ and $j = \max\{k \in \{1, \ldots, n\} : q_k \in M\}$. Consider the order $<'$ obtained from $<$ by reversing the order between the elements in $\{q_i, \ldots, q_j\}$:

$$q_1 <' q_2 <' \ldots <' q_{i-1} <' q_j <' q_{j-1} <' \ldots <' q_i <' q_{j+1} <' q_{j+2} <' \ldots <' q_n.$$

We assert that $<'$ is a compatible order. Indeed, let $q_x <' q_y <' q_z$, and assume that they are not in the same order as in $<$. Hence $x < i \leq y < z \leq j$ (or symmetrically $i \leq x < y \leq j < z$). Then $q_x < q_i \leq q_z < q_y \leq q_j$, from which we get $d(q_x, q_i) \leq d(q_x, q_z) \leq d(q_x, q_y) \leq d(q_x, q_j) = d(q_x, q_i)$, and then $d(q_y, q_z) \leq d(q_x, q_z) = d(q_x, q_y)$, proving the compatibility of $<'$.

Since (X,d) is flat, $<$ and $<'$ are either equal or reverse to each other. In the first case, this means that $i = j$ and M is trivial. In the second case, this means that $i = 1$ and $j = n$. Suppose now that there are $\alpha < \beta$ in $\{2,\dots,n-1\}$ with $q_\alpha, q_\beta \notin M$. Then $d(q_1,q_\alpha) \leq d(q_1,q_\beta) = d(q_\beta,q_n) \leq d(q_\alpha,q_n) = d(q_1,q_\alpha)$, implying that those quantities are all equal to the same value δ. From this, $d(u,v) = \delta$ for each $u \in M$, $v \notin M$, hence $X \setminus M$ is also an mmodule, not containing q_1 and q_n. By the previous analysis, $X \setminus M$ is a trivial module, hence $|M| = n-1$.

Consequently, we conclude that any non-trivial mmodule of (X,d) is of the form $X \setminus \{q_i\}$ for some $i \in \{2,\dots,n-1\}$. Suppose that (X,d) admits two non-trivial mmodules $X \setminus \{q_i\}$ and $X \setminus \{q_j\}$ with $1 < i < j < n$ (notice that we need $n \geq 4$). Then for all $x \in X \setminus \{q_i,q_j\}$ we have $d(x,q_i) = d(q_i,q_j) = d(q_j,x)$, hence $\{q_i,q_j\}$ is an mmodule. Since $n > 3$, this is a contradiction to the fact that the non-trivial mmodules have cardinality $n-1$. This proves that M is unique.

Finally, let $\Delta = \mathrm{diam}(X) = d(q_1,q_n)$ be the diameter of (X,d), let j be such that $M = X \setminus \{q_j\}$. Suppose that q_j is the end of a diametral pair, that is $d(q_i,q_j) = \Delta$ for any $q_i \in M$. Then for all $i \in \{1,\dots,j-1\}$ and all $k \in \{j+1,\dots,n\}$, we have $\Delta = d(q_i,q_j) \leq d(q_i,q_k) \leq d(q_1,q_n) = \Delta$, implying that $\{q_1,\dots,q_j\}$ is a non-trivial mmodule not containing q_n, a contradiction. ∎

4.2. Conical Robinson spaces. In this subsection, we consider the conical Robinson spaces (X,d), i.e., the Robinson spaces having a cotrivial copoint partition $\mathcal{C}_p = \{\{p\}, X \setminus \{p\}\}$. Let $d(p,x) = \delta$ for any $x \in X \setminus \{p\}$. Let also $X' = X \setminus \{p\}$. We will show how to compute, from any compatible order $<'$ of (X',d), a compatible order $<$ of (X,d).

Let $<'$ be a compatible order of (X',d). Let x_* and x^* be respectively the minimal and maximal points of $<'$. By a *hole* of $<'$ we will mean any pair (x,y) of consecutive points $x,y \in X'$ of $<'$ with $x <' y$. We will also consider the pair (x^*,x_*) as a hole (this corresponds to turning $<'$ into a circular order). For a hole $(x,y) \neq (x^*,x_*)$, let $<_{(x,y)}$ be the total order obtained by inserting p in the hole (x,y), i.e., by setting $u <_{(x,y)} v$ when $u <' v$ if $u,v \in X'$, and $u <_{(x,y)} p$, $p <_{(x,y)} v$ for any $u,v \in X'$ such that $u \leq' x$ and $y \leq' v$. If $(x,y) = (x^*,x_*)$, then we set $v <_{(x,y)} p$ for all $v \in X'$ (p is located to the right of x^*) or we set $p <_{(x,y)} u$ for all $u \in X'$ (p is located to the left of x_*). We will call a hole (x,y) of $<'$ *admissible* if $<_{(x,y)}$ is a compatible order of (X,d).

LEMMA 4.2. *Let $<'$ be a compatible order of (X',d). A hole $(x,y) \neq (x^*,x_*)$ of $<'$ is admissible if and only if for any $u,v \in X'$ with $u <' v$, the following conditions hold:*

(1) *$d(u,v) \geq \delta$ if $u \leq' x$ and $y \leq' v$;*
(2) *$d(u,v) \leq \delta$ if $v \leq' x$ or $y \leq' u$.*

Proof. Consider a hole $(x,y) \neq (x^*,x_*)$. Pick any three points $u,v,w \in X$ such that $u <_{(x,y)} v <_{(x,y)} w$. If $p \notin \{u,v,w\}$, then $u <' v <' w$ and thus $d(u,w) \geq \max\{d(u,v),d(v,w)\}$. Now, let $p \in \{u,v,w\}$. First suppose that $p = u$ (the case $p = w$ is similar). Then $d(u,v) = d(u,w) = \delta$. Consequently, $d(v,w) \leq d(u,w)$ if and only if $d(v,w) \leq \delta$, i.e., condition (2) holds. Now suppose that $p = v$. Then $d(u,v) = d(v,w) = \delta$. Then $d(u,w) \geq \max\{d(u,v),d(v,w)\}$ if and only if $d(u,w) \geq \delta$, i.e., condition (1) holds. Consequently, the hole (x,y) is admissible if and only if both conditions (1) and (2) hold. ∎

LEMMA 4.3. *Let $<'$ be a compatible order of (X',d). The hole (x^*,x_*) is admissible if and only if $d(x_*,x^*) \leq \delta$. Moreover in that case, a hole $(x,y) \neq (x^*,x_*)$ is admissible if and only if $d(x,y) = \delta$.*

Proof. Since $<'$ is a compatible order of (X',d) and for any $u,v \in X'$ we have $x_* \leq' u <' v \leq' x^*$ or $x_* \leq' v <' u \leq' x^*$, we conclude that $d(x_*,x^*) \geq d(u,v)$. Therefore (x^*,x_*) is admissible if and only if $\delta \geq d(x_*,x^*)$. The second part of the lemma is a consequence of Lemma 4.2, as condition (2) is implied by the fact that the diameter is δ. ∎

A pair $\{u,v\}$ of X' is called *large* if $d(u,v) > \delta$. In view of Lemma 4.3, we can suppose that $d(x_*,x^*) > \delta$, thus X' contains at least one large pair. An interval $[a,b]$ of $<'$ is *nontrivial* if $a <' b$.

LEMMA 4.4. *If (X,d) is a conical Robinson space with apex p containing at least one large pair, and $<'$ is a compatible order on (X',d), then the intersection of the intervals of $<'$ defined by all large pairs of (X',d') is a nontrivial interval $[a,b]$ of $<'$.*

Proof. Let $\{u_1,v_1\}$ and $\{u_2,v_2\}$ be two large pairs with intervals whose intersection is not nontrivial. We may assume that $u_1 <' v_1 \leq' u_2 <' v_2$. Then $d(v_2,u_1) \geq d(v_2,v_1) \geq d(v_2,u_2) > \delta$. Let $<$ be a compatible order on (X,d), and denote $\{x,y,z\} = \{u_1,v_1,v_2\}$ such that $x < y < z$. Then $d(x,p) = d(p,y) = \delta < d(x,y)$ implies $x < p < y$, and similarly $y < p < z$, contradiction.

Let $\mathcal{F}$ be the family of intervals defined by large pairs. By the *Helly property* for intervals of a total order, if the intervals pairwise intersect, then they all intersect. Moreover, if the pairwise intersections are nontrivial, then the intersection is also nontrivial: indeed, the common intersection is precisely the intersection of the interval of $\mathcal{F}$ with the minimal rightmost end, and of the interval of $\mathcal{F}$ with the maximal leftmost end. Consequently, the intersection $[a,b]$ of all intervals of $<'$ defined by large pairs is a nontrivial interval. □

PROPOSITION 4.5. *If (X,d) is a conical Robinson space with apex p, then any compatible order $<'$ of (X',d) has at least one admissible hole included in $[a,b]$. Furthermore, any hole (x,y) of $<'$ located in $[a,b]$ and such that $d(x,y) \geq \delta$ is admissible.*

Proof. If $d(x_*,x^*) \leq \delta$, the assertion follows from Lemma 4.2. So, suppose that $<'$ contains at least one large pair. Let (y,z) be a hole with $a \leq' y <' z \leq' b$ and $d(y,z) \geq \delta$. Then for any w,w' with $w \leq' y$, $z \leq' w'$, $d(w,w') \geq \delta$, fulfilling condition (1) in Lemma 4.2. Moreover, since $a < z$, $d(z,x^*) \leq \delta$, hence for any $u <' v$ with $z <' u$, $d(u,v) \leq d(z,x^*) \leq \delta$. By a symmetric argument on x_* and y, condition (2) is also fulfilled, yielding that (y,z) is admissible. Finally, such a hole exists: let $<$ be a compatible order for (X,d), and let (y,z) be a hole in $[a,b]$ such that $y < p$ and $p < z$. Such a hole exists because $a < p$ and $p < b$. Then $y < p < z$ and $d(y,p) = \delta$ implies $d(y,z) \geq \delta$. □

By Proposition 4.5, an admissible hole of $<'$ can be computed in $O(n)$ time. First we compute the intersection $[a,b]$ of all intervals of $<'$ defined by large pairs, test all holes located in $[a,b]$, and find one (x,y) such that $d(x,y) \geq \delta$. To compute $[a,b]$, we use the following lemma:

LEMMA 4.6. *If (X,d) is a conical Robinson space with apex p and $<'$ is a compatible order of (X',d) with minimal element x_* and maximal element x^*, then b coincides with the leftmost point b' of $<'$ such that $d(x_*,b') > \delta$ and a coincides with the rightmost point a' of $<'$ such that $d(a',x^*) > \delta$.*

Proof. Since $\{x_*,b'\}$ and $\{a',x^*\}$ are large pairs, by Lemma 4.4 $[a,b] \subseteq [x_*,b'] \cap [a',x^*]$. Since this intersection is a nontrivial interval, $a' < b'$ and $[x_*,b'] \cap [a',x^*] = [a',b']$. If $a \neq a'$, then there exists a large pair $\{x,y\}$ such that $a' <' x <' y \leq x^*$. From the definition of a' it follows that $d(x,x^*) \leq \delta$. Since $d(x,y) > \delta$ and $x <' y \leq x^*$, this contradicts that $<'$ is a compatible order. Thus $a = a'$ and analogously, $b = b'$. □

5. Mmodules in Robinson spaces. In this section, we investigate the mmodules and the copoint partitions in Robinson spaces.

5.1. Copoints in Robinson spaces. Let $\mathcal{C}_p = \{C_0 = \{p\}, C_1, \ldots, C_k\}$ be a copoint partition with attaching point p of a Robinson space (X,d). For a copoint C_i, denote by δ_i the distance $d(p,x)$ for any point $x \in C_i$ and we suppose that $i < j$ implies that $\delta_i \leq \delta_j$. For $C_i, C_j \in \mathcal{C}_p, i \neq j$, let δ_{ij} be the distance $d(x,y)$ between any two points $x \in C_i$ and $y \in C_j$. Since $\mathcal{C}_p$ is a stable partition, δ_{ij} is well-defined, moreover δ_{ij} coincides with $\widehat{d}(C_i,C_j)$ in the quotient space $(\mathcal{C}_p,\widehat{d})$. In this subsection, we investigate how in a compatible order $<$ of (X,d) the copoints of $\mathcal{C}_p$ compares to the point p and which copoints of $\mathcal{C}_p$ are not comparable to p.

Notice that each subspace $(C_i \cup \{p\}, d), i = 1,\ldots,k$ is a cone over (C_i,d) with apex p. Applying Proposition 4.5 to the Robinson space (C_i,d) and to the restriction $<_i$ of $<$ to C_i, we know that $<_i$ admits at least one admissible hole. If this hole is defined by the rightmost and the leftmost points of $<_i$, then C_i is not divided in two parts, otherwise p divides C_i in two parts C_i^l and C_i^r. The next result shows that C_i^l and C_i^r are not only intervals of $<_i$ but also of the global compatible order $<$:

LEMMA 5.1. *If $<$ is a compatible order of (X,d), then any copoint $C_i \in \mathcal{C}_p$ defines at most two intervals C_i^l and C_i^r of $<$. If C_i defines two intervals C_i^l, C_i^r, then $C_i^l < p < C_i^r$ or $C_i^r < p < C_i^l$.*

Proof. Let $C_i^r := \{u \in C_i : p < u\}$ contains at least 2 points and let $x,z \in C_i^r$ be its minimum and maximum elements, respectively. Let $y \in X$ with $x < y < z$, and let $w \in X \setminus C_i$ such that either $w < x$ or $z < w$. If $w < x$, then $d(w,x) \leq d(w,y) \leq d(w,z) = d(w,x)$, where the last equality comes from the fact that C_i is an mmodule that contains x,z but not w. Similarly, if $z < w$, then $d(w,z) \leq d(w,y) \leq d(w,x) = d(w,z)$. Hence for any such w, the distance $d(w,y)$ is constant for any $y \in [x,z]$. As C_i is a maximal mmodule not containing p, this implies that $[x,z] \subseteq C_i$, that is C_i^r is an interval (possibly empty). Symmetrically,

$C_i^l := \{u \in C_i : u < p\}$ is also an interval. □

A copoint $C_i \in \mathcal{C}_p$ is *separable* if $\operatorname{diam}(C_i) > \delta_i$, *non-separable* if $\operatorname{diam}(C_i) < \delta_i$, and *tight* if $\operatorname{diam}(C_i) = \delta_i$.

LEMMA 5.2. *If $<$ is a compatible order of (X, d), then any separable copoint C_i defines two intervals C_i^l and C_i^r of $<$ such that $d(x, y) \geq \delta_i$ for any $x \in C_i^l, y \in C_i^r$ and $\operatorname{diam}(C_i^l) \leq \delta_i$, $\operatorname{diam}(C_i^r) \leq \delta_i$.*

Proof. Let u, v be a diametral pair of C_i, with $d(u, v) = \operatorname{diam}(C_i) > \delta_i$. Then necessarily either $u < p < v$ or $v < p < u$, otherwise it would contradict the compatibility of $<$. Thus $u \in C_i^l$ and $v \in C_i^r$. By Lemma 5.1, C_i^l and C_i^r are intervals of $<$. Let $x, y \in C_i$ with $x \in C_i^l$ and $y \in C_i^r$, then $x < p < y$, whence $d(x, y) \geq \max\{d(x, p), d(y, p)\} = \delta_i$. Let $x, y \in C_i$ with $x, y \in C_i^l$ or $x, y \in C_i^r$, say the first, then $d(x, y) \leq \max\{d(x, p), d(y, p)\} = \delta_i$. □

LEMMA 5.3. *If $<$ is a compatible order of (X, d), then any non-separable copoint C_i defines a single interval of $<$.*

Proof. If $x < p < y$ for two points $x, y \in C_i$, then $d(x, y) \geq \max\{d(x, p), d(y, p)\} = \delta_i$, which is impossible because $d(x, y) \leq \operatorname{diam}(C_i) < \delta_i$. Thus C_i must be located on one side of p. By Lemma 5.1, C_i is an interval of $<$. □

LEMMA 5.4. *If $<$ is a compatible order of (X, d) and C', C'' are two intervals of $<$ defined by a tight copoint C_i such that $C' < p < C''$, then the order $<'$ defined by the rule:*

$$\text{for any } u < v, \text{ if } C' < u < C'' \text{ and } v \in C'', \text{ then } v <' u, \text{ otherwise } u <' v,$$

is a compatible order of (X, d). Consequently, if (X, d) is Robinson, then there exists a compatible order for which each tight copoint is a single interval.

The order $<'$ is thus obtained from $<$ by moving C'' immediately after C'. By symmetry, one could get a similar result by moving instead C' in front of C''.

Proof. Pick any three points $x, y, z \in X$ such that they are not identically ordered by $<$ and by $<'$. Then we can suppose without loss of generality that $x \in C''$ and $C' < y < C''$. Notice also that $C' \cup C''$ is an interval of $<'$. We will show now that whatever is the order of x, y, z with respect to $<$, it does not yield a contradiction for $<'$. First, let $z \in X \setminus C_i$. Let x_ℓ be any point of C'. Since $C' \cup C''$ is an interval of $<'$ containing x, x_ℓ and not containing y, z, the order of x, y, z along $<'$ is the same as the order of x_ℓ, y, z along $<'$, which is the same as the order of x_ℓ, y, z along $<$. Since $d(x, z) = d(x_\ell, z), d(x, y) = d(x_\ell, y)$ and $<$ is a compatible order, the result follows.

Now, let $z \in C''$. Then we can suppose that $y < x < z$ (the other case $y < z < x$ is similar) and thus $d(y, z) \geq \max\{d(y, x), d(x, z)\}$. In this case, we have $x <' z <' y$. As $d(y, z) = d(y, x)$, we obtain $d(y, x) \geq \max\{d(y, z), d(x, z)\}$ and we are done. Finally, let $z \in C'$. Then we have $z < y < x$. If $p < y$ or $p = y$ (the case $y < p$ is symmetric), then since C_i is a tight copoint, we obtain $\delta = d(z, p) \leq d(z, y) = d(y, x) \leq d(z, x) = \delta$. Consequently, $d(y, x) = d(y, z) = d(x, z)$ and thus the triple x, y, z yields no contradiction for $<'$. □

Lemmas 5.2 to 5.4 imply the following result:

PROPOSITION 5.5. *If (X, d) is a Robinson space and $\mathcal{C}_p$ is a copoint partition of X, then there exists a compatible order $<$ in which each copoint C_i with $\operatorname{diam}(C_i) \leq \delta_i$ is an interval of $<$ located either to the left or to the right of p and each copoint C_i with $\operatorname{diam}(C_i) > \delta_i$ defines two intervals C_i^l and C_i^r of $<$ such that $C_i^l < p < C_i^r$.*

Next we consider only compatible orders of (X, d) satisfying the conditions of Proposition 5.5.

5.2. Compatible orders from compatible orders of copoints and extended quotient. Let $\mathcal{C}_p = \{C_0 = \{p\}, C_1, \ldots, C_k\}$ be a copoint partition with attaching point p of a Robinson space (X, d). For a separable copoint C_i an *admissible bipartition* is a partition C_i into C_i' and C_i'' such that $\operatorname{diam}(C_i') \leq \delta_i, \operatorname{diam}(C_i'') \leq \delta_i$, and $d(x, y) \geq \delta_i$ for any $x \in C_i'$ and $y \in C_i''$. This partition is defined by applying Proposition 4.5 to each $(C_i, d), i = 1, \ldots, k$ and to each conical Robinson space $(C_i \cup \{p\}, d)$. Notice that C_i' and C_i'' are no longer mmodules because the distances between the points of C_i' and C_i'' are not necessarily the same. We will call C_i' and C_i'' *halved copoints*.

Let $\mathcal{C}_p^*$ denote the set of all non-separable and tight copoints of $\mathcal{C}_p$ plus the set of all halved copoints corresponding to admissible bipartitions of separable copoints. The *extended quotient* of (X,d) is the dissimilarity space $(\mathcal{C}_p^*, d^*)$ defined in the following way: for $i \neq j$, the distance $d^*(\alpha,\beta)$ between a pair of copoints or halved copoints α,β, (1) is δ_{ij} when one is indexed by i and the other is indexed by j, and (2) is $\mathrm{diam}(C_i)$, the diameter of C_i, when α and β are the two half copoints from the same copoint C_i. Notice that for any points $u \in \alpha$ and $v \in \beta$, in the first case we have $d^*(\alpha,\beta) = d(u,v)$ and in the second case, we have $d^*(\alpha,\beta) \geq d(u,v)$.

Lemma 5.6. *If (X,d) is a Robinson space, then for any $p \in X$, its quotient $(\mathcal{C}_p, \widehat{d})$ and its extended quotient $(\mathcal{C}_p^*, d^*)$ are Robinson spaces.*

Proof. It suffices to isometrically embed $(\mathcal{C}_p, \widehat{d})$ in $(\mathcal{C}_p^*, d^*)$ and $(\mathcal{C}_p^*, d^*)$ in (X,d). The map $\widehat{\varphi} : \mathcal{C}_p \to \mathcal{C}_p^*$ maps any non-separable or tight copoint C_i to itself and any separable copoint C_i to one of its halves. From the definitions of $(\mathcal{C}_p, \widehat{d})$ and $(\mathcal{C}_p^*, d^*)$, $\widehat{\varphi}$ is an isometric embedding. The map $\varphi^* : \mathcal{C}_p^* \to X$ is defined as follows. We select one point x_i in each non-separable or tight copoint C_i and set $\varphi^*(C_i) = x_i$ and we select a diametral pair $\{x_i', x_i''\}$ for each separable copoint C_i separated into C_i' and C_i'' and set $\varphi^*(C_i') = x_i', \varphi^*(C_i'') = x_i''$. From the definition of $(\mathcal{C}_p^*, d^*)$, φ^* is an isometric embedding. ◻

Now, we will show that if (X,d) is a Robinson space, then from any compatible order $<^*$ of the extended quotient $(\mathcal{C}_p^*, d^*)$ and from the compatible orders $<_i$ of the copoints (C_i, d) of $\mathcal{C}_p$, we can define a compatible order $<$ of (X,d). We assume that any compatible order $<_i$ of a separable copoint C_i defines an admissible bipartition $\{C_i', C_i''\}$ of C_i as defined in Subsection 4.2. The total order $<$ is defined as follows: for two points x,y of X we set $x < y$ if and only if (1) $x \in \alpha, y \in \beta$ for two different points $\alpha,\beta \in \mathcal{C}_p^*$ and $\alpha <^* \beta$ or (2) $x,y \in \alpha \in \mathcal{C}_p^*$, $\alpha \subseteq C_i$, and $x <_i y$.

Proposition 5.7. *If (X,d) is a Robinson space, then $<$ is a compatible order of (X,d).*

Proof. Pick any three distinct points $u < v < w$ of X and let $u \in \alpha, v \in \beta$, and $w \in \gamma$ with $\alpha,\beta,\gamma \in \mathcal{C}_p^*$. If $\alpha = \beta = \gamma \subseteq C_i$, then $u <_i v <_i w$ and the result follows from the fact that $<_i$ is a compatible order of (C_i, d). Now, let α,β, and γ be distinct. Then $\alpha <^* \beta <^* \gamma$ and $d^*(\alpha,\gamma) \geq \max\{d^*(\alpha,\beta), d^*(\beta,\gamma)\}$. Since $d^*(\alpha,\gamma) \geq d(u,w)$, $d^*(\alpha,\beta) \geq d(u,v)$, and $d^*(\beta,\gamma) \geq d(v,w)$, the inequality $d(u,w) \geq \max\{d(u,v), d(v,w)\}$ holds if $d^*(\alpha,\gamma) = d(u,w)$. Now suppose that say $d^*(\alpha,\beta) > d(u,v)$. From the definition of d^* this implies that α and γ are the halved copoints C_i' and C_i'' of C_i and say $C_i' <^* \beta <^* p <^* C_i''$. Then β belongs to a copoint C_j with $\delta_j \leq \delta_i$. Since $u,w \in C_i$ and $v \in C_j$, $d(u,v) = d(v,w) = \delta_{ij} = d^*(\beta,\gamma) = d^*(\alpha,\beta) \leq \delta_i$. On the other hand, since $\{C_i^l, C_i^r\}$ is an admissible partition of C_i, $d(u,w) \geq \delta_i$ and we are done.

Finally, let $\alpha = \beta$ or $\beta = \gamma$, say the first. First suppose that α and γ are halved copoints of C_i: $\alpha = C_i^l$ and $\gamma = C_i^r$. Then $u <_i v <_i w$ and since $<_i$ is a compatible order of (C_i, d), we are done. Now suppose that α and γ belong to different copoints, say $\alpha \subseteq C_i$ and $\gamma \subseteq C_j$. Since $u,v \in C_i$ and $w \in C_j$, $d(u,w) = d(v,w) = \delta_{ij}$. It remains to prove that $d(u,v) \leq \delta_{ij}$. By Proposition 5.5, there is a compatible order for which α is an interval, in particular for which w is not between u and v. This implies $d(u,v) \leq \max\{d(u,w), d(v,w)\} = \delta_{ij}$. This concludes the proof of the proposition. ◻

6. Proximity orders. In this section, we show how to compute a compatible order of a p-trivial Robinson space and of an extended quotient of any Robinson space. The algorithmic details and the complexity analysis of these results are deferred to Section 7. Our main tool is the concept of p-proximity order.

6.1. p-Proximity orders. Let (X,d) be a Robinson space with a compatible order $<$ and let p be a point of X. A *p-proximity order* for $<$ is a total order $\prec$ on X such that

(PO1) for all distinct $x,y \in X$, if $x \prec y$, then $d(p,x) \leq d(p,y)$,

(PO2) for all distinct $x,y \in X \setminus \{p\}$, $x \prec y$ implies that either $y < p$ and $y < x$, or $y > p$ and $y > x$ (*ie.* y is not between p and x in the compatible order $<$).

Notice that p is the minimum of $\prec$. Given the compatible order $<$, a p-proximity order for $<$ can be built by shuffling the elements smaller than p in reverse order into the elements larger than p, and adding p as the minimum. The following lemma is a restatement of (PO2).

Lemma 6.1. *If $\prec$ is a p-proximity order for a compatible order $<$, then for any point $u \in X$, the set $\{x \in X : x \prec u\}$ is an interval of $<$.*

A *p-proximity pre-order* for a compatible order $<$ is a pre-order $\preceq$ on X which can be refined to a p-

proximity order for $<$, and two distinct elements are equal only if they belong to the same copoint of $\mathcal{C}_p$. One can build a p-proximity pre-order for a compatible order $<$ without the knowledge of $<$:

PROPOSITION 6.2. *Let (X,d) be a Robinson space and let p be any point of X. Then one can compute a p-proximity pre-order $\prec$ for some (unknown) compatible order $<$ of (X,d) in time $O(n^2)$.*

Proof. We use the partition refinement algorithm, which can be described as iterated steps. Recall that each step consists in refining some part S of the current partition $\mathcal{P}$ into parts $S_1,\dots S_k$, using a pivot $q \notin S$, where each S_i is an equivalence class of S for the relation $x \sim y$ if and only if $d(q,x) = d(q,y)$. We only study the steps for which $k > 1$, and our indices are chosen such that for any $x \in S_i$, $y \in S_j$, $d(q,x) < d(q,y)$ iff $i < j$. We will denote $d(q,S_i)$ the distance $d(q,s)$ for any $s \in S_i$ and take an arbitrary compatible order $<$. We may assume by symmetry that $p < q$.

We start with the pre-order $\preceq$ where $p \precneq x = y$ for all $x,y \in X \setminus \{p\}$. We use the invariant that *the equality classes of $\preceq$ are the part of the current refinement $\mathcal{P}$.* The algorithm terminates when the current partition $\mathcal{P}$ consists only of copoints, which implies that the equality classes of $\preceq$ will be the copoints, as desired. We also keep the invariant that *if $x \precneq y$, then (PO1) and (PO2) are satisfied for the pair x,y, that is $d(p,x) \leq d(p,y)$ and y cannot be between p and x.*

Consider the first refinement step. Then $S = X \setminus \{p\}$ and $q = p$. We refine $\preceq$ by setting $x \precneq y$ for any $x \in S_i$ and $y \in S_j$ with $i < j$. In this case we have $d(p,x) < d(p,y)$, hence (PO1) and (PO2) are satisfied for this pair. Consider now any other refinement step from S with pivot q such that $S \preceq q$. Let α be the minimum index such that $d(q,S_\alpha) > d(p,q)$ (we may take $\alpha = k+1$). We refine $\preceq$ by setting $x \precneq y$ when $x \in S_i$, $y \in S_j$ and either $j < i < \alpha$, or $\alpha \leq i < j$ or $i < \alpha \leq j$ holds. That is, we have $S_{\alpha-1} \precneq S_{\alpha-2} \precneq \dots S_1 \precneq S_\alpha \precneq S_{\alpha+1} \precneq \dots \precneq S_k$. Notice that $d(p,s)$ is constant for all $s \in S$, so (PO1) is satisfied for any $x \in S_i$, $y \in S_j$. We assert that (PO2) is also satisfied. Indeed, let $x \in S_i$, $y \in S_j$ with $S_i \precneq S_j$. We distinguish three cases. If $j < i < \alpha$, then $d(q,y) < d(q,x) \leq d(q,p)$. Since $S \precneq q$, $x < y < q$, this implies that y is not between p and x. If $i < \alpha$, $j \geq \alpha$, then $d(q,x) < d(q,p) \leq d(q,y)$. Together with $S \precneq q$ this implies $y < p < x < q$, whence y is not between p and x. Finally, if $\alpha \leq i < j$, then $d(q,p) \leq d(q,x) < d(q,y)$. Again together with $S \precneq q$, we get $y < x < p < q$ or $y < p < x < q$, whence y is not between p and x. Finally consider a refinement step such that $q \precneq S$. We refine $\preceq$ by setting $x \precneq y$ when $x \in S_i$, $y \in S_j$ and $i < j$. Since $q \precneq S$, we have that x and y are not between p and q. If y is between p and x, then either $p < q < y < x$ or $x < y < p < q$. But both cases contradict $d(q,x) < d(q,y)$.

This constructive proof leads to Algorithm 7.1 for constructing a p-proximity pre-order. Its correctness and complexity is provided by Lemmas 7.1 to 7.3. ∎

For p-trivial Robinson spaces, the p-proximity pre-order is an order:

PROPOSITION 6.3. *Let (X,d) be a p-trivial Robinson space. Then a p-proximity pre-order is an order and can be computed in time $O(n^2)$.*

Proof. Since all copoints of $\mathcal{C}_p$ are trivial, a p-proximity pre-order $\preceq$ is by definition a total order, hence $\preceq$ is a p-proximity order. Hence the result follows from Proposition 6.2. ∎

6.2. p-Proximity orders for extended quotients. Let (X,d) be a Robinson space and $p \in X$. By Lemma 3.6, $(\mathcal{C}_p,\widehat{d})$ is p-trivial. By Lemma 5.6, $(\mathcal{C}_p,\widehat{d})$ and $(\mathcal{C}_p^*,d^*)$ are Robinson, but $(\mathcal{C}_p^*,d^*)$ is not p-trivial. Nevertheless, $(\mathcal{C}_p^*,d^*)$ has a p-proximity order:

PROPOSITION 6.4. *For an extended quotient $(\mathcal{C}_p^*,d^*)$ of a Robinson space (X,d) with k copoints one can compute a p-proximity order $\prec$ for some (unknown) compatible order $<$ of $(\mathcal{C}_p^*,d^*)$ in $O(k^2)$.*

Proof. By Lemma 5.6, $(\mathcal{C}_p^*,d^*)$ is Robinson. By Proposition 6.2 there is a p-proximity pre-order $\preceq$ for a compatible order $<$ of $(\mathcal{C}_p^*,d^*)$. One can easily see that the copoints of p in $(\mathcal{C}_p^*,d^*)$ are either trivial or of the form (C_i',C_i'') for a separable copoint $C_i \in \mathcal{C}_p$. We refine $\preceq$ into an order $\prec$ by arbitrarily ordering each such pair (C_i',C_i''). We assert that there exists a compatible order $<^*$ on $\mathcal{C}_p^*$ having $\prec$ as a p-proximity order. By Lemma 5.6, $(\mathcal{C}_p^*,d^*)$ is isometric to the restriction of (X,d) to the following set: the point p, one representative x_i for each tight or non-separable copoint C_i, and a diametral pair (x_i',x_i'') as representatives for each separable copoint.

We define $<^*$ from $<$ by transposing the representatives (x_i',x_i'') of each separable copoint C_i with $x_i'' < x_i'$, so that $x_i' <^* x_i''$. Then, $<^*$ is also a compatible order: indeed the dissimilarity matrices with rows and columns ordered by $<$ and $<^*$ are identical, because $\{x_i', x_i''\}$ is an mmodule of $\mathcal{C}^*$. Next we prove that $\prec$ is a p-proximity order for $<^*$. To prove (PO1), as $\prec$ is a refinement of $\preceq$, we only need to prove that $d(p, x_i') \le d(p, x_i'')$ for any separable copoint. This is indeed so because $d(p, x_i') = d(p, x_i'')$. To prove (PO2), we first prove that $\preceq$ is a pre-order for $<^*$.

We distinguish four cases. First, suppose that $x \precneqq y$ and neither x nor y was transposed. Then the relative order of p, x and y does not change, hence in $<^*$, y is not between p and x. Second, suppose that $x \precneqq y$, x was transposed as member of a pair (x_i', x_i''), and y was not transposed. This means that $x_i'' < p < x_i' < y$ (because $\preceq$ is a p-proximity pre-order), and after transposing we have $x_i' <^* p <^* x_i'' <^* y$, hence y is not between x_i' and p nor between p and x_i'' in $<^*$. Now, suppose that $x \precneqq y$ and x was not transposed, but y was transposed as member of a pair (y_j', y_j''). This is similar to the previous case: $y_j'' < p < x < y_j'$ or $y_j'' < x < p < y_j'$, and after transposition, we have $y_j' <^* p <^* x <^* y_j''$ or $y_j' <^* x <^* p <^* y_j''$. Anyways we get that p is not between p and x in $<^*$. Finally, suppose that $x \precneqq y$ and both x, y were transposed. Hence they belong to pairs (x_i', x_i'') and (y_j', y_j''), respectively. Then $y_j'' < x_i'' < p < x_i' < y_j'$ and $y_j' <^* x_i' <^* p <^* x_i'' <^* y_j''$. Hence y is not between p and x in $<^*$. Consequently, we proved that $\preceq$ is a pre-order for $<^*$.

Since $\prec$ is a refinement of $\preceq$, it suffices to prove (PO2) for the pairs (x_i', x_i''). But $x_i' <^* p <^* x_i''$, hence x_i'' is not between p and x_i', concluding the proof that $\prec$ is a p-proximity order for $<^*$. ∎

6.3. Compatible orders from p-proximity orders. By definition of a p-proximity order $\prec$ for some compatible order $<$ of (X, d), if we know $\prec$ and we know which elements of $X \setminus \{p\}$ are located to the left and to the right of p in $<$, then we can recover the compatible order $<$ from $\prec$:

Lemma 6.5. *Let (X, d) be a Robinson space with a compatible order $<$. Then $<$ is fully determined by a p-proximity order for $<$ and the bipartition $(L, R) = (\{x \in X : x < p\}, \{x \in X : x > p\})$. More precisely, for any $u, v \in X \setminus \{p\}$, we have $u < v$ if and only if*
- *either $u \in L$, $v \in R$,*
- *or $u, v \in L$, and $v \prec u$,*
- *or $u, v \in R$ and $u \prec v$.*

The following result shows how to partition $X \setminus \{p\}$ into the sets L and R:

Proposition 6.6. *Let (X, d) be a Robinson dissimilarity and $\prec$ be a p-proximity order on (X, d). Then one can build in time $O(n^2)$ the sets $L = \{u \in X : u < p\}$ and $R = \{u \in X : u > p\}$ for some compatible order $<$ for which $\prec$ is a p-proximity order.*

Proof. Let $m \in X$ be the maximum element of $\prec$. We construct a function side $: X \setminus \{p\} \to \{L, R\}$ in such a way that $v \in \text{side}(v)$. To this end, we define a graph $G = (X \setminus \{p\}, \{uv : u \prec v \wedge d(u, v) \ne d(p, v)\}$. Pick any edge uv of G. Then either $d(u, v) < d(p, v)$, implying that side(u) = side(v), or $d(u, v) > d(p, v)$, implying that side$(u) \ne$ side(v). As a consequence, if G contains only one connected component, then the map side is well-defined, and so are the sets L and R (up to symmetry). We now assume that G has at least two connected components and let $\mathcal{K}$ be the set of connected components of G. Let $K_m \in \mathcal{K}$ be the component of G containing m. Consider the following graph $H = (\mathcal{K}, \{CC' : C \text{ and } C' \text{ are not comparable by } \prec\})$.

Claim 1. *Either the graph H is connected or there is a point $u \ne m$ and a connected component $\mathcal{K}'$ of H such that $M = \{x \in X : x \preceq u\}$ is an mmodule and $\bigcup \mathcal{K}' = \{x \in X : u \prec x\} = \overline{M}$.*

Proof. Suppose that H is not connected and let u be the maximum of $\prec$ among all points of $X \setminus \{p\}$ outside the component K_m. Let $M = \{v \in X : v \preceq u\}$. By the choice of u and the definition of H, $X \setminus M$ is induced by a connected component $\mathcal{K}'$ of H. Let $x, y \in M$ and $q \notin M$. Then $x \prec q$ and $y \prec q$. Since x and y are not in the connected component containing q, we also have $d(p, q) = d(q, x)$ and $d(p, q) = d(q, y)$, hence $d(q, x) = d(q, y)$, proving that M is an mmodule. ∎

If the graph H is connected, then we set $M = \{p\}$, $\overline{M} = X \setminus \{p\}$. Otherwise, if C and C' are adjacent in $\mathcal{K}'$, then (up to symmetry), there are $y \in C$, $x, z \in C'$ with $x \prec y \prec z$ and $xz \in E$. This implies that $d(p, x) \le d(p, y) \le d(p, z)$ and $d(x, z) \ne d(p, z)$. Since y is not in the same component of G as x, z, we also have $d(p, y) = d(x, y)$ and $d(p, z) = d(y, z)$. We consider two cases:

Case 1: side(x) = side(z), that is $d(x, z) < d(p, z)$. Then either $d(y, z) \le d(x, z)$, implying side(y) = side(z), or $d(y, z) \ge d(p, z)$, implying side$(y) \ne$ side(z). The case $d(x, z) < d(y, z) < d(p, z)$ is not possible, as then we would have that y is between p and x, contradicting $x \prec y$.

Case 2: side$(x) \neq$ side(z), that is $d(x,z) > d(p,z) = d(y,z)$. Since $x \prec y$, y is not between x and p, hence side(y) = side(z).

In all cases, the sides of elements of C are determined by the sides of those of C'. Since $\overline{M}$ is induced by a connected component $\mathcal{K}'$ of H, the side of each element in $\overline{M}$ is determined by the choice of the side for m. Let $(L_{\overline{M}}, R_{\overline{M}})$ be one of the two possible bipartitions of $\overline{M}$. Let $<_{\overline{M}}$ be the compatible order of $\{p\} \cup \overline{M}$ obtained from $(L_{\overline{M}}, R_{\overline{M}})$ and the restriction of $\prec$ to $\{p\} \cup \overline{M}$.

If $M = \{p\}$, let $<_M$ be the trivial order on M, and $(L_M, R_M) = (\varnothing, \varnothing)$. Otherwise, using the restriction of $\prec$ to M, by induction we can find a bipartition L_M, R_M of $M \backslash \{p\}$ for some compatible order $<_M$ on M. Notice that by Lemma 6.5, $<_M$ is determined by $\prec_M$ and (L_M, R_M). Finally, set $(L,R) = (L_{\overline{M}} \cup L_M, R_M \cup R_{\overline{M}})$, and let $<$ be the order on X given by

$$x < y \text{ if } \begin{cases} & x, y \in \overline{M}, x <_{\overline{M}} y, \\ \text{or} & x, y \in M, x <_M y, \\ \text{or} & x \in M, y \in \overline{M}, p <_{\overline{M}} y, \\ \text{or} & x \in \overline{M}, y \in M, x <_{\overline{M}} p. \end{cases}$$

We assert that $<$ is a compatible order, establishing that (L,R) is a bipartition defining a compatible order. It can be readily checked that $<$ is a total order. Let $u < v < w$ and we must prove that $d(u,w) \geq \max\{d(u,v), d(v,w)\}$. We proceed by cases.

If $\{u,v,w\} \subseteq M$, or $\{u,v,w\} \subseteq \overline{M}$, then the result follows since $<_M$ and $<_{\overline{M}}$ are compatible orders on M and $\overline{M} \cup \{p\}$. Now, let $u, v \in M$, $w \in \overline{M}$. Since M is an mmodule, $d(u,w) = d(v,w)$. Since $u \prec w$ and $v \prec w$, w is not between p and u nor between p and v in $<$, hence w is not between u and v. Thus $d(u,v) \leq \max\{d(u,w), d(v,w)\} = d(u,w)$. The case $u \in \overline{M}$, $v, w \in M$ is symmetric. The case $u, w \in M$, $v \in \overline{M}$ is impossible, as $u < v$ implies $v \in R_{\overline{M}}$ and $v < w$ implies $v \in L_{\overline{M}}$. Now, let $u \in M$, $v, w \in \overline{M}$. Then $p <_{\overline{M}} v <_{\overline{M}} w$, yielding $\max\{d(p,v), d(v,w)\} \leq d(p,w)$, and since M is an mmodule containing p and u, we obtain $\max\{d(u,v), d(v,w)\} \leq d(u,w)$. The case $u, v \in \overline{M}$, $w \in M$ is symmetric. Finally, let $u, w \in \overline{M}$, $v \in M$. Then $u \in L_{\overline{M}}$ and $w \in R_{\overline{M}}$, thus $u <_{\overline{M}} p <_{\overline{M}} w$. As M is an mmodule containing v and p, this implies that $\max\{d(u,v), d(v,w)\} = \max\{d(u,p), d(p,w)\} \leq d(u,w)$. Consequently, $<$ is a compatible order of (X,d).

This constructive proof leads to Algorithm 7.4 for constructing a compatible order. Its correctness and complexity is provided by Lemma 7.5. ∎

Consequently, we obtain the following result:

PROPOSITION 6.7. *Let (X,d) be p-trivial or flat Robinson space on n points. Then a compatible order on X can be computed in $O(n^2)$ time. Analogously, if $(\mathcal{C}_p^*, d^*)$ is an extended quotient of a Robinson space (X,d) with k copoints, then a compatible order on $\mathcal{C}_p^*$ can be computed in $O(k^2)$.*

Proof. If (X,d) is p-trivial, then the result follows from Propositions 6.3 and 6.6 and Lemma 6.5. Now suppose that (X,d) is flat and let p be a diametral point of (X,d). By Proposition 4.1, (X,d) is p-trivial, thus we can apply the previous case. Finally, if $(\mathcal{C}_p^*, d^*)$ is an extended quotient of a Robinson space (X,d), then C_p^* contains at most $2k$ points. Consequently the result follows from Propositions 6.4 and 6.6 and Lemma 6.5. ∎

7. A divide-and-conquer algorithm. The results of Sections 3, 5, and 6 (namely, Proposition 5.7 and Proposition 6.7) lead to the following algorithm for computing a compatible order of a Robinson space (X,d):

1. Compute a copoint partition $\mathcal{C}_p$ of (X,d),
2. Recursively find a compatible order $<_i$ for each copoint C_i of $\mathcal{C}_p$,
3. Classify the copoints of $\mathcal{C}_p$ into separable, tight, and non-separable and construct the extended quotient $(\mathcal{C}_p^*, d^*)$ of (X,d),
4. Compute a p-proximity order $\prec$ for the extended quotient $(\mathcal{C}_p^*, d^*)$,
5. Build a compatible order $<^*$ for $(\mathcal{C}_p^*, d^*)$ using $\prec$,
6. Merge the compatible order $<^*$ on $\mathcal{C}_p^*$ with the compatible orders $<_i$ on the copoints C_i of $\mathcal{C}_p$ to get a total order $<$ on X, using Proposition 5.7,
7. If $<$ is not a compatible order of (X,d), then return "not Robinson", otherwise return $<$.

In this section, we detail the algorithm and prove that its complexity is $O(n^2)$. The algorithm does not use fancy data structures besides persistent lists and balanced binary search trees.

7.1. Data structures. We use a bracketed notation for *persistent lists* [29], with $[]$ being the empty list. We introduce the two operators $\cdot$ and $+\!\!+$ defined by

$$x \cdot [l_1, \dots, l_n] = [x, l_1, \dots, l_n]$$
$$[l_1, \dots, l_n] +\!\!+ [l'_1, \dots, l'_m] = [l_1, \dots, l_n, l'_1, \dots, l'_m].$$

One can implement the operator $\cdot$ in $O(1)$ time and $+\!\!+$ with $O(n)$ time (where n is the length of the left operand), using single-linked lists. Extracting the first element of a list also takes $O(1)$ time. We will also use the reverse operation with time-complexity $O(n)$, and concatenate with time-complexity $\sum_{i=1}^{k-1} |l_i|$, where

$$\text{reverse}([l_1, l_2, \dots, l_n]) = [l_n, l_{n-1}, \dots, l_1]$$
$$\text{concatenate}(l_1, \dots, l_k) = l_1 +\!\!+ \dots +\!\!+ l_k.$$

Balanced binary search trees (see e.g. [29]) are used solely to sort in increasing order a list of n elements with at most k distinct key values, in time $O(n \log k)$. This is achieved by building a balanced binary search tree of the key values appearing in the list, each associated to a list of elements sharing that key value. The sorting algorithm consists in inserting each element in the list associated to its key value, then concatenating all the associated lists in increasing order of the key values. Each insertion takes time $O(\log k)$, and the final concatenation takes time $O(n)$. We denote the binary search tree operation by $\text{insert}(T, key, value)$ (inserts a value with a given comparable key), $\text{containsKey}(T, key)$ (checks whether there is a value with a given key), $\text{get}(T, key)$ (retrieves the value associated to a given key), and $\text{values}(T)$ (returns the list of keys in increasing order of their values).

Finally, to represent the extended quotient $(\mathcal{C}_p^*, d^*)$, we select a set of representatives of tight, non-separable, and halved copoints: the point p, one representative x_i for each tight or non-separable copoint C_i, and a diametral pair (x'_i, x''_i) for each separable copoint.

7.2. Refinements and recursive refinements. The first ingredient of the algorithm is a refinement algorithm to compute the copoints of $\mathcal{C}_p$. It differs from the classical refinement algorithm in that aspect that we additionally want the copoints to be sorted in increasing p-proximity pre-order. The Algorithm 7.1, based on Proposition 6.2 uses Algorithm 7.2 as a subroutine, which is in charge of refining a set S relatively to a single external pivot $q \notin S$.

Algorithm 7.1 $\text{recursiveRefine}(p, In, S, Out)$

Input: a Robinson space (X, d) (implicit), a point $p \in X$, a set $S \subseteq X$, two disjoint subsets $In, Out \subseteq X \setminus S$ of inner-pivots and outer-pivots.
Output: an ordered partition of S (encoding a partial p-proximity pre-order).

```
if In ∪ Out = ∅ then
    return [S]
let q ∈ In ∪ Out                                   ▷ choose q to be the first element of In or Out
let [S_1, ..., S_m] = refine(q, S)
if q ∈ Out then
    let α = min({j ∈ {1, ..., m} : d(S_α, q) > d(p, q)} ∪ {m + 1})
    let [S'_1, ..., S'_m] = [S_{α-1}, S_{α-2}, ..., S_1, S_α, S_{α+1}, ..., S_m]
else
    let [S'_1, ..., S'_m] = [S_1, ..., S_m]
for i ∈ {1, ..., m} do
    let In_i = concatenate(S'_1, ..., S'_{i-1}, In \ {q})
    let Out_i = concatenate(S'_{i+1}, ..., S'_k, Out \ {q})
    let T_i = recursiveRefine(p, In_i, S'_i, Out_i)
return concatenate(T_1, ..., T_m)
```

LEMMA 7.1. *Algorithm 7.2 outputs a partition $\mathcal{S} = (S_1, \dots, S_m)$ of S in time $O(|S| \log m)$, where*

Algorithm 7.2 refine(q, S)

Input: a Robinson space (X, d) (implicit), a point $q \in X$, a subset $S \subseteq X$.
Output: an ordered partition of S, by increasing distance from q

```
let T be an empty balanced binary tree, with keys in ℕ
for x ∈ S do
    if ¬containsKey(T, d(q, x)) then
        insert(T, d(q, x), [])
    insert(T, d(q, x), x · get(T, d(q, x)))
return values(T)
```

(i) for each $1 \le i \le m$*, for all* $x, y \in S_i$*,* $d(q, x) = d(q, y)$*,*
(ii) for each $1 \le i < j \le m$*, for all* $x \in S_i, y \in S_j$*,* $d(q, x) < d(q, y)$*.*

Proof. First, $\mathcal{S}$ is a partition, since each element is inserted in a list of T exactly once. Each class of $\mathcal{S}$ is at constant distance from q since we use the distances to q as keys. Finally, the classes of $\mathcal{S}$ are sorted by increasing distances from q, because values(T) returns its associated values in increasing order of keys. The complexity analysis follows from the fact that the binary search tree contains at most m keys, hence each of its elementary operations are in $O(\log m)$. The evaluation of values(T) can be done in $O(m)$ operations by a simple right-to-left DFS traversal of the binary search tree, inserting (not appending) each list in T from farthest to closest into the returned list. ∎

Lemma 7.2. *Algorithm 7.1 called on* $(p, [p], S, [])$*, outputs a partition of* S *into copoints attached to* p*, given in increasing* p*-proximity pre-order.*

Proof. The algorithm closely follows the proof of Proposition 6.2, with the difference that it uses two lists *In* and *Out* instead of a set of future pivots. To process correctly the pivots q for any given S_i, we must know whether q is smaller or greater than the elements in S'. To this end, instead of keeping, for each set S_i, a set $Z(S_i)$ of elements to be used as pivot for S', we use two different sets In_i and Out_i, where In_i contains pivots smaller than the elements of S_i, and Out_i contains pivots greater then the elements of S_i, relative to the p-proximity pre-order. This is possible since the call to *refine* returns a partition in increasing order of distances from q, and depending on whether $q \in In$ or $q \in Out$, we can determine the pre-order $S'_1 \prec S'_2 \prec \ldots \prec S'_k$ of the classes. ∎

We will compute the complexity of Algorithm 7.1 without counting the time spent by all recursive calls to *refine*, which will be done at a later stage in our analysis.

Lemma 7.3. *Without counting the time spent in the calls to Algorithm 7.2, Algorithm 7.1 with inputs* $(p, [q], S, [])$ *runs in time* $O(|S|^2 - \sum_{i=1}^{k^\star} |S_i^\star|^2)$*, where its output is* $[S_1^\star, \ldots, S_{k^\star}^\star]$*.*

Proof. The complexity depends on $|S|$ and $|In \cup Out|$ and is dominated by the computations of In_i, Out_i, and T_i for all $i \in \{1, \ldots, k\}$. Indeed, computing α and reversing a prefix of the list take $O(m)$ operations, and the concatenation on the last line takes $O(k^\star)$ operations, but the construction of In_i and Out_i takes $\Omega(n)$ operations. Thus we only analyze the latter.

For each elements $x, y \in S$, during all the recursive calls, y will be added and removed at most once into a list In_i, Out_i associated to a set S_i containing x. Notice that this is true only when using persistent lists for L and R, allowing to share $In \setminus \{q\}$ and $Out \setminus \{q\}$ without copying the lists. It follows that the number of such insertions is bounded by $\sum_{i=1}^{k} |C_i|\,|S \setminus C_i| = |S|^2 - \sum_{i=1}^{k} |C_i|^2$. Also the total number of recursive calls that do not terminate on the first two lines must be bounded by this same quantity; indeed each of those recursive calls remove one element in one of these lists. ∎

7.3. Separation of separable copoints. Algorithm 7.3 partitions separable copoints into halves.

Lemma 7.4. *Algorithm 7.3 returns a pair* (x, C_i) *with* $x \in C_i$ *if it is not separable, or two pairs* $[(x', C'_i), (x'', C''_i)]$ *where* $x' \in C'_i$*,* $x'' \in C''_i$*,* (x', x'') *is a diametral pair of* C_i *and* C'_i, C''_i *are the halved copoints of* C_i *if it is separable. It runs in* $O(|C_i|)$*-time.*

Proof. Since C_i is a copoint, $(\{p\} \cup C_i, d)$ is a conical Robinson space, and Proposition 4.5 applies. Thus when the diameter Δ of C_i exceeds $\delta = d(p, C_i)$, Algorithm 7.3 looks for a hole (x, y) in $[a, b]$ with

Algorithm 7.3 separateIfSeparable(p, C_i)

Input: a Robinson space (X, d) (implicit), a point $p \in X$, a copoint C_i attached to p, sorted in a compatible order $<$.
Output: a copoint or two halved copoints, depending on whether C_i is separable, each with a representative.

```
let x_*, x^* be the first and last elements of C_i
let Δ = d(x_*, x^*), δ = d(p, x_*)
if Δ ≤ δ then
    return [(x_*, C_i)]
for x ∈ C_i in increasing compatible order do
    let y be the element consecutive to x
    if d(x_*, x) ≤ δ and d(y, x^*) ≤ δ and d(x, y) ≥ δ then
        return [(x_*, {u ∈ C_i : u ≤ x}), (x^*, {u ∈ C_i : u ≥ y})]
```

$d(x, y) \geq \delta$, hence with $d(x_*, x) \leq \delta$ and $d(y, x^*) \leq \delta$. Otherwise, C_i is not separable and (x^*, x_*) is a hole. The complexity of the algorithm can be readily checked. ∎

Remark 2. *Algorithm 7.3 may not return a valid partition when C_i is not a copoint of a Robinson space, because the condition tested by the second **if** statement may never be satisfied. Thus when testing whether a dissimilarity space is Robinson, if the space is not Robinson, the algorithm may either fail in this procedure, or return an order that is not compatible.*

7.4. Finding a compatible order from a proximity order. Algorithm 7.4 implements the ideas of the proof of Proposition 6.6 to compute a compatible order from a p-proximity order. Proposition 6.6 gives a bipartition L, R which together with Lemma 6.5 provides a compatible order. Algorithm 7.4 builds the bipartition L, R directly.

Algorithm 7.4 sortByBipartition(p, S)

Input: A Robinson space (X, d) (implicit), a point $p \in X$, a subset $S \subseteq X$ in p-proximity order.
Output: $S \cup \{p\}$ in a compatible order.

```
let L = [], R = [], Undecided = reverse(S)
for q ∈ S in decreasing order do
    if q = first(Undecided) then
        choose arbitrarily: either R ← q · R or L ← q · L
        Undecided ← Undecided \ {q}
    let Skipped = []
    for x ∈ Undecided from first to last do
        if d(x, q) = d(p, q) then
            Skipped ← x · Skipped
            continue
        if d(x, q) < d(p, q) ⟺ q ∈ R then
            R ← x · R
            L ← Skipped ++ L
        else
            L ← x · L
            R ← Skipped ++ R
        Skipped ← []
    Undecided ← reverse(Skipped)
return reverse(L) ++ [p] ++ R
```

Lemma 7.5. *Algorithm 7.4 returns a compatible order of $\{p\} \cup S$ in $O(|S|^2)$ time.*

Proof. Let G and H be the two graphs introduced in the proof of Proposition 6.6. We first prove that at the end, the lists L, R define a bipartition of S. First we show that each element is in L or R. Initially, *Undecided* contains each element in decreasing p-proximity order. Notice the invariant that

$\{L, R, \textit{Undecided}\}$ is a partition of S at the start and end of each iteration. Indeed, during the iteration, each element of *Undecided* is inserted in L, in R, or in *Skipped*, and elements are removed from *Skipped* only to be inserted in L or R. We also have the following invariant during the outer loop on q: *Undecided* contains only elements smaller than or equal to q. This invariant is easily checked since the first instruction in the outer loop removes q if it is still there. Together, the two invariants ensure that when exiting the outer loop, $\{L, R\}$ is a bipartition of S.

Now, we show that when an element is inserted in L or R in the inner loop, it is done correctly. When inserting x to L or R, this is correct because of an edge in the graph G. When appending *Skipped* to L or R, this is also correct, because x and q are in the same component C of G, and all elements skipped are in components incomparable with C, which correspond to edges of H.

Then, we prove that when q is inserted in L or R, this happens because $M = \{x \in S : x \preceq q\}$ is an mmodule. Indeed, all elements $y \in S \setminus M$ have been processed in previous iteration of the outer loop. As seen in the proof of Proposition 6.6, in that case the elements of *Undecided* are M and we can choose an arbitrary side for q. Finally, observe that the order of elements in L and R are reversed relative to the order in *Undecided*, that is they are in increasing p-proximity order. Thus the **return** instruction is correct in reversing L and not reversing R.

The complexity of Algorithm 7.4 easily follows as each loop iterates at most $|S|$ times and observing that the complexity of appending *Skipped* is amortized over the insertions into *Skipped*. ∎

7.5. The divide-and-conquer algorithm. Now, we are ready to describe and analyze the divide-and-conquer algorithm for recognizing Robinson dissimilarities.

Algorithm 7.5 findCompatibleOrder(X)

Input: a Robinson space (X, d) (d is implicit).
Output: a compatible order for X (as a sorted list).

```
if X is empty then
    return []
let p ∈ X, X' = X \ {p}
let [C_1, ..., C_k] = recursiveRefine(p, [p], X', [])
let representedCopoints = []
for i ∈ {1, ..., k} in decreasing order do
    let C'_i = findCompatibleOrder(C_i)
    representedCopoints ← separateIfSeparable(p, C'_i) ++ representedCopoints
let [(x_1, T_1), ..., (x_k', T_k')] = representedCopoints
let [x_σ(1), ..., p, ..., x_σ(k')] = sortByBipartition(p, [x_1, ..., x_k'])
return concatenate(T_σ(1), ..., [p], ..., T_σ(k'))
```

We will use the following auxiliary result:

LEMMA 7.6. *If $T : \mathbb{N} \to \mathbb{N}$ satisfies the recurrence relation $T(n) \leq \sum_{i=1}^{k} T(n_i) + n \log k$, for all partitions $\sum_{i=1}^{k} n_i = n$ of n in $k \geq 2$ positive integers, then $T(n) = O(n^2)$.*

Proof. By convexity of the function $\sum_{i=1}^{k} x_i^2$, the maximum of $\sum_{i=1}^{k} n_i^2$ over all partitions of n in k parts is attained by a partition with one class with $n - k + 1$ points and $k - 1$ singletons, and has value $(n - k + 1)^2 + (k - 1)$. Assume that for all $p < n$, $T(p) \leq \alpha p^2$ for some $\alpha \geq \frac{1}{2}$. Then

$$\begin{aligned}
T(n) &\leq \sum_{i=1}^{k} \alpha n_i^2 + n \log k \\
&\leq \alpha(n - k + 1)^2 + \alpha(k - 1) + n \log k \\
&= \alpha n^2 - 2\alpha(k - 1)(n - k) + n \log k \\
&\leq \alpha n^2 - n(2\alpha(k - 1) - \log k),
\end{aligned}$$

where the last inequality follows from $n-k \leq n$. It suffices to prove that $2\alpha(k-1)-\log k$ is positive, which is true because $\alpha \geq \frac{1}{2}$ and $k \geq 2$. ∎

We continue with the main result of the paper.

THEOREM 7.7. *Algorithm 7.5 computes a compatible order of a Robinson space (X,d) in $O(n^2)$ time.*

Proof. The correction follows from Proposition 5.7 that proves that a compatible order on (X,d) can be built by composing a compatible order on each copoint or halved copoint with a compatible order on the extended quotient space (which exists by Proposition 5.7). By Lemma 7.4 and by induction, each T_i is a tight or non-separable copoint or a halved copoint in increasing compatible order, with representative x_i, and by Lemma 7.5, σ sorts the representatives $(x_i)_{i\in\{1,\dots,k\}}$ and p in a compatible order, so that Proposition 5.7 applies.

We analyze the complexity of Algorithm 7.5 by counting separately the number of operations done in the procedures *refine*, *recursiveRefine* and *sortByBipartition*. All the other operations can be done in linear-time at each level of recursion, thus in $O(|X|^2)$ times in total.

- *recursiveRefine* contributes $O(|X|^2)$ in the total complexity; indeed, applying Lemma 7.3, the first call takes $\alpha\cdot|X|^2-\sum_{i=1}^{k}|C_i|^2$ (for some constant α), while the cost of *recursiveRefine* in the recursive calls are at most $\alpha\sum_{i=1}^{k}|C_i|^2$ by induction, summing to $\alpha|C|^2$.
- *refine* contributes $O(|C|^2)$ in the total complexity, because it follows the recurrence relation described in Lemma 7.6.
- *sortByBipartition* contributes $O(|X|^2)$ in the total complexity; indeed, considering the recursion tree of calls to *findCompatibleOrder*, one can see that each call to *sortByBipartition* uses $O(k^2)$ operations where k is the arity of the corresponding node. Hence the contribution of *sortByBipartition* is of the form $\beta\sum_{i=1}^{l}k_i^2$ where l is the number of nodes and k_i the arity of the k_ith node. But, because each node, inner or leaf, can be associated to a unique element in X, $\sum_{i=1}^{l}k_i = |X|-1$, implying that $\beta\sum_{i=1}^{l}k_i^2 \leq \beta n^2$ by convexity.

Summing up all the contributions, we get that *findCompatibleOrder* runs in time $O(|X|^2)$. ∎

REMARK 3. *Algorithm 7.5 can be transformed into a recognition algorithm by simply testing in $O(|X|^2)$ time if the returned sorted list is a compatible order on (X,d). If this is not the case, from the results of previous sections it follows that (X,d) is not a Robinson space.*

REMARK 4. *If (X,d) is p-trivial, then all copoints C_i have size 1, thus in this case the Algorithm 7.5 is no longer recursively applied to the copoints. In particular, if (X,d) is a flat Robinson space and p is diametral, then by Proposition 4.1 (X,d) is p-trivial.*

8. Conclusion. In this paper, we investigated the structure of mmodules and copoint partitions in dissimilarity spaces, and, more particularly, in Robinson spaces. Based on these results, we presented a divide-and-conquer algorithm for recognizing Robinson matrices in optimal $O(n^2)$ time. In the companion paper [7], we also establish a correspondence between the mmodular decomposition of a Robinson dissimilarity and its PQ-tree. PQ-trees are used to encode all compatible orders of a Robinson space and using one such compatible order (say, computed by our algorithm), we will show how to construct the PQ-tree. We hope that the approach used in this paper can be used to recognize more general classes of dissimilarities, in particular tree-Robinson and circular-Robinson dissimilarities.

As we already mentioned, the first recognition algorithm of Robinson spaces running in optimal $O(n^2)$ time was presented by Préa and Fortin [30] and uses PQ-trees as a data structure for encoding all compatible orders. Due to this, even if optimal, this algorithm is not simple. PQ-trees are also used in the general case of the Atkins et al. [3] recognition algorithm based on the spectral approach and in the recent algorithm of [2] for strict circular seriation. Our optimal recognition algorithm is simple and was relatively easy to implement in OCaml, as it should also be in any mainstream programming language. Since it uses only basic data structures, it can be casted as practical. The program solves seemingly hard instances on 1000 points in half a second, and instances on 10000 points in less than a minute on a standard laptop. Among the different random generators we used to evaluate our program, the hardest instances were obtained by shuffling Robinson Toeplitz matrices with coefficients in $\{0,1,2\}$.

REFERENCES

[1] J. Aracena and C. Thraves Caro, *On the recognition of strong-Robinsonian incomplete matrices*, arXiv:2101.03033, (2021).
[2] S. Armstrong, C. Guzmán, and C. A. Sing Long, *An optimal algorithm for strict circular seriation*, SIAM J. Math. Data Science, 3 (2021), pp. 1223–1250.
[3] J. E. Atkins, E. G. Boman, and B. Hendrickson, *A spectral algorithm for seriation and the consecutive ones problem*, SIAM J. Comput., 28 (1998), pp. 297–310.
[4] J.-P. Barthélemy and F. Brucker, *NP-hard approximation problems in overlapping clustering*, J. Classif., 18 (2001), pp. 159–183.
[5] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J. Comput. Syst. Sci., 13 (1976), pp. 335–379.
[6] M. J. Brusco, *A branch-and-bound algorithm for fitting anti-Robinson structures to symmetric dissimilarity matrices*, Psychometrika, 67 (2002), pp. 459–471.
[7] M. Carmona, V. Chepoi, G. Naves, and P. Préa, *Modules and PQ-tees in Robinson spaces*, 2022. in preparation.
[8] V. Chepoi and B. Fichet, *Recognition of Robinsonian dissimilarities*, J. Classif., 14 (1997), pp. 311–325.
[9] V. Chepoi, B. Fichet, and M. Seston, *Seriation in the presence of errors: NP-hardness of l_∞-fitting Robinson structures to dissimilarity matrices*, J. Classif., 26 (2009), pp. 279–296.
[10] V. Chepoi and M. Seston, *Seriation in the presence of errors: A factor 16 approximation algorithm for l_∞-fitting Robinson structures to distances*, Algorithmica, 59 (2011), pp. 521–568.
[11] F. Critchley and B. Fichet, *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, Springer, 1994, pp. 5–65.
[12] E. Diday, *Orders and overlapping clusters in pyramids.*, in Multidimensional Data Analysis, J. e. a. de Leeuw, ed., DSWO Press, Leiden, 1986.
[13] C. Durand and B. Fichet, *One-to-one correspondences in pyramidal representation: A unified approach.*, in Conf. of IFCS, 1987, pp. 85–90.
[14] A. Ehrenfeucht, H. N. Gabow, R. M. McConnell, and S. Sullivan, *An $O(n^2)$ divide-and-conquer algorithm for the prime tree decomposition of two-structures and modular decomposition of graphs*, J. Algorithms, 16 (1994), pp. 283–294.
[15] A. Ehrenfeucht and G. Rozenberg, *Theory of 2-structures, part I: Clans, basic subclasses, and morphisms*, Theor. Comput. Sci., 70 (1990), pp. 277–303.
[16] A. Ehrenfeucht and G. Rozenberg, *Theory of 2-structures, part II: Representation through labeled tree families*, Theor. Comput. Sci., 70 (1990), pp. 305–342.
[17] F. Fogel, A. d'Aspremont, and M. Vojnovic, *Serialrank: Spectral ranking using seriation*, NeuroIPS, 27 (2014).
[18] M. Ghandehari and J. Janssen, *An optimization parameter for seriation of noisy data*, SIAM J. Discr. Math., 33 (2019), pp. 712–730.
[19] M. Habib and C. Paul, *A survey of the algorithmic aspects of modular decomposition*, Comput. Sci. Review, 4 (2010), pp. 41–59.
[20] L. Hubert, *Some applications of graph theory and related non-metric techniques to problems of approximate seriation: The case of symmetric proximity measures*, British J. Math. Stat. Psychol., 27 (1974), pp. 133–153.
[21] L. Hubert, P. Arabie, and J. Meulman, *Graph-theoretic representations for proximity matrices through strongly-anti-Robinson or circular strongly-anti-Robinson matrices*, Psychometrika, 63 (1998), pp. 341–358.
[22] L. Hubert, P. Arabie, and J. Meulman, *The structural representation of proximity matrices with MATLAB*, ASA-SIAM Series on Statistics and Applied Probability, SIAM, 2006.
[23] Y. Issartel, C. Giraud, and N. Verzelen, *Optimal embedding on the sphere in non-parametric latent space models*, 2021.
[24] R. E. Jamison, *Copoints in antimatroids*, Congr. Numer, 29 (1980), pp. 535–544.
[25] M. Laurent and M. Seminaroti, *A Lex-BFS-based recognition algorithm for Robinsonian matrices*, Discr. Appl. Math., 222 (2017), pp. 151–165.
[26] M. Laurent and M. Seminaroti, *Similarity-first search: a new algorithm with application to Robinsonian matrix recognition*, SIAM J. Discr. Math., 31 (2017), pp. 1765–1800.
[27] M. Laurent, M. Seminaroti, and S. Tanigawa, *A structural characterization for certifying Robinsonian matrices*, Electr. J. Combin., 24 (2017), p. P2.21.
[28] B. Mirkin and S. Rodin, *Graphs and Genes*, Springer, 1984.
[29] C. Okasaki, *Purely Functional Data Structures*, Cambridge University Press, 1999.
[30] P. Préa and D. Fortin, *An optimal algorithm to recognize Robinsonian dissimilarities*, J. Classif., 31 (2014), pp. 351–385.
[31] A. Recanati, T. Kerdreux, and A. d'Aspremont, *Reconstructing latent orderings by spectral clustering*, arXiv preprint arXiv:1807.07122, (2018).
[32] W. S. Robinson, *A method for chronologically ordering archaeological deposits*, Amer. Antiq., 16 (1951), pp. 293–301.
[33] M. Seston, *Dissimilarités de Robinson: algorithmes de reconnaissance et d'approximation*, PhD thesis, Université de la Méditerranée, Aix-Marseille 2, 2008.
[34] M. Seston, *A simple algorithm to recognize Robinsonian dissimilarities*, COMPSTAT 2008, (2008), pp. 241–248.
[35] M. L. van De Vel, *Theory of Convex Structures*, Elsevier, 1993.