Journal of Classification 34:191-222 (2017)
DOI: 10.1007/s00357-017-9230-1

Robinsonian Matrices: Recognition Challenges

D. Fortin

INRIA, Paris, France

Abstract: Ultrametric inequality is involved in different operations on (dis)similarity matrices. Its coupling with a compatible ordering leads to nice interpretations in seriation problems. We accurately review the interval graph recognition problem for its tight connection with recognizing a dense Robinsonian dissimilarity (precisely, in the anti-ultrametric case). Since real life matrices are prone to errors or missing entries, we address the sparse case and make progress towards recognizing sparse Robinsonian dissimilarities with lexicographic breadth first search. The ultrametric inequality is considered from the same graph point of view and the intimate connection between cocomparability graph and dense Robinsonian similarity is established. The current trend in recognizing special graph structures is examined in regard to multiple lexicographic search sweeps. Teaching examples illustrate the issues addressed for both dense and sparse symmetric matrices.

Keywords: Robinsonian matrices; Interval graphs; Cocomparability graphs; Lexicographic searches; Partition refinement.

1. Introduction

Since the classic seriation problem introduced by Robinson (Robinson, 1951), for chronological dating, the Robinson property has been used in many classification applications and other fields. A symmetric matrix fulfills the Robinson property if its entries monotonically vary away from the diagonal along the rows and the columns; precisely, if its entries decrease

The author is specially grateful to M. Habib and P. Préa for numerous discussions about the relationships between interval graphs, LexBFS and bottleneck inequalities on three and four points. An anonymous referee deeply interacts with the manuscript so that it brings many clarifications and less imperfections; many thanks to him/her for the deep and fruitful reviewing.

Corresponding Author's Address: D. Fortin, INRIA, Domaine de Voluceau 78153 Le Chesnay cedex, France, email: inria@dominique.fortin.io.

Published online: 24 June 2017

(resp. increase) in similarity (resp. dissimilarity). In spite of the fact that the property looks very much the same in either case, we focus on the dissimilarity case since it is easier to solve. A (symmetric) $n \times n$ dissimilarity matrix $\mathbf{D}$ is Robinsonian if and only if there exists an ordering of its rows (and columns consequently) such that

$$\max(d_{ij}, d_{jk}) \leq d_{ik}, \quad \text{for all } 1 \leq i < j < k \leq n$$

using $d_{ij} = \mathbf{D}[i, j]$ for ease of reading. The recognition problem amounts to finding one (possibly all) such ordering, leading to a Robinsonian matrix which is a starting point for rich interpretations in different fields. It is worth beginning with the following observations: if all entries are the same, the matrix is clearly Robinsonian and if all entries are different then it suffices to sort a single line and to check that the corresponding permutation fits the Robinson property; in both cases, complexity remains $O(n^2)$.

Despite the fact that many authors contributed to the Robinsonian matrix recognition problem, we should stress a few, but crucial, steps:

- V. Chepoi and B. Fichet (Chepoi and Fichet, 1997) divide and conquer the problem through a sequence of decreasing threshold balls in $O(n^3)$ time complexity; each threshold separates items between inside, outside and on the sphere itself. In spite of well ordered items between inside and outside items, the sphere items may be either on one side or on the other side leading to some ambiguities,
- D. Fortin and P. Préa (Préa and Fortin, 2014) consider the sequence of increasing threshold to profit from well known results on clique trees and interval graph property of each threshold graph (Booth and Lueker, 1976). A clever trick relies on merging PQ-trees which describe the structure of every threshold graph, then on checking whether any permutation stored by the final PQ-tree, fits the Robinson property. It achieves the optimal complexity $O(n^2)$. A PQ$-tree$ remembers permutation sets with P$-node$ for a symmetric subgroup and Q$-node$ for a subgroup associated with a cycle; we refer to the seminal paper by Booth and Luekker for a detailed description of PQ$-trees$ and their properties,
- M. Laurent and M. Seminaroti (Laurent and Seminaroti, 2015a) observe more accurately that dense matrices should fit the unit interval (**UI** for short) ordering instead of the interval (**I** for short) ordering to yield a divide and conquer (in the increasing framework) algorithm based on the recognition of **UI** interval graphs (Corneil, 2004). It achieves a worst case $O(n^2 \log n)$ complexity and in practice much better behavior since it depends on the number of different entries. It

outperforms, to our feeling, all PQ-tree based alternatives and is remarkable in the sense that given a permutation leading to the Robinson property, all compatible permutations (in other words the PQ-tree itself) may be constructed a posteriori.

Some remarks are in order at this point:

- Without loss of generality, all matrices are assumed to have *non-negative* entries.
- The dense character of the matrix plays a crucial role; let us consider a sparse dissimilarity with missing domino entries (2 consecutive entries in either dimension),

$$\begin{bmatrix} 0 & & & 1 & 3 & 3 & 3 \\ & 0 & & 1 & 3 & 3 & 3 \\ & & 0 & 1 & & & 2 \\ 1 & 1 & 1 & 0 & 2 & 2 & 2 \\ 3 & 3 & & 2 & 0 & & \\ 3 & 3 & & 2 & & 0 & \\ 3 & 3 & 2 & 2 & & & 0 \end{bmatrix}$$

an exhaustive search for permutations and completions leads to

filling	permuted	Robinson cycles
$\begin{bmatrix}1 & 1\end{bmatrix}$		$(2,3)$
$\begin{bmatrix}1 & 2\end{bmatrix}$		$(2,3)$
$\begin{bmatrix}2 & 1\end{bmatrix}$		$(4,5)(2,3)$
$\begin{bmatrix}2 & 2\end{bmatrix}$		identity

$$\begin{bmatrix} 0 & & 1 & & 3 & 3 & 3 \\ & 0 & 1 & & 3 & 3 & 3 \\ 1 & 1 & 0 & 1 & 2 & 2 & 2 \\ & & 1 & 0 & 1 & 1 & 2 \\ 3 & 3 & 2 & 1 & 0 & & \\ 3 & 3 & 2 & 1 & & 0 & \\ 3 & 3 & 2 & 2 & & & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & & 1 & & 3 & 3 & 3 \\ & 0 & 1 & & 3 & 3 & 3 \\ 1 & 1 & 0 & 1 & 2 & 2 & 2 \\ & & 1 & 0 & 1 & 2 & 2 \\ 3 & 3 & 2 & 1 & 0 & & \\ 3 & 3 & 2 & 2 & & 0 & \\ 3 & 3 & 2 & 2 & & & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & & & 1 & 3 & 3 & 3 \\ & 0 & & 1 & 3 & 3 & 3 \\ & & 0 & 1 & 2 & 2 & 2 \\ 1 & 1 & 1 & 0 & 2 & 2 & 2 \\ 3 & 3 & 2 & 2 & 0 & & \\ 3 & 3 & 2 & 2 & & 0 & \\ 3 & 3 & 2 & 2 & & & 0 \end{bmatrix}$$

Moreover, the threshold graph at level 1, exhibits a claw subgraph (a vertex adjacent to three vertices, not adjacent among themselves) so that the unit interval condition is violated.

- On the one hand, (Chepoi, Fichet, and Seston, 2009) have shown that completion is NP-hard; on the other hand, practical applications are prone to both errors and missing entries. Moreover, it is likely that for a sampling matrix, mostly all permutations fail the Robinsonian property; on the opposite, for human optimized transportation processes, some examples are known to exhibit an underlying Monge property, i.e., the linear version (Dietrich, 1990; Fortin and Rudolf, 1998) instead of the bottleneck case we consider; see also (Burkard et al., 1998; Çela, Deineko, and Woeginger, 2012; Laurent and Seminaroti, 2015b) of the relationship.

- About the underlying convexity; let us consider two convex functions f, g then $\max(f, g)$ is convex too. It is likely that Robinson himself noticed this fact and introduced a *pseudo* identity operator that transforms discrete entry $[i, j]$ into a linear (hence convex) function so that Robinson property reads $\max(d_{ij}, d_{jk}) - d_{ik} \leq 0, \quad$ for all $1 \leq i < j < k \leq n$ which turn out to be convex constraints under the *linear* transform. It becomes an easy convex optimization problem and a posteriori explains the optimality of recognition, in dense case. What about the reverse inequality? It reads $\max(d_{ij}, d_{jk}) - d_{ik} \geq 0, \quad$ for all $1 \leq i < j < k \leq n$ and reveals a more difficult constraints structure, namely reverse convex constraints. A necessary and sufficient condition to check feasibility is to maximize the minimum over all the convex left hand sides and to verify that the maximum is nonnegative; however, $\min(f_i)$ for convex functions f_i is a piecewise convex function (see Tsevendorj, 2001 for optimality conditions). Ultrametrics, equivalently bottleneck triangle inequality $\max(d_{ij}, d_{jk}) \leq d_{ik}$, play a prominent role in classification too and are well studied (Dahlhaus, 1993); however, as an optimization problem in the above sense, it remains widely open to our knowledge and is quoted as a challenging issue in terms of recognition of cocomparability graphs, taken as the underlying threshold graphs associated with the matrix.

Therefore, both practical and theoretical issues remain to be addressed. The article aims at the Robinsonian (anti-ultrametrics) matrix recognition but intends to make some progress towards the anti-Robinsonian (ultrametrics) matrix recognition as well. Since it relies on convexity and lexicographic searches, in Section 2 we recall the fundamental results for lexicographic graph searches and prove them again in the setting of partition refinement (Habib et al., 2000) instead of usual labeling schemes; despite the fact that it does not bring anything new, it collects in a concise and coherent way, the ambiguities occurring in the *pivot* part and are at the heart of all crucial points mentioned above. In Section 3, we revisit the four LexBFS (lexicographic breadth first search) sweeps algorithm for interval graph recognition, under the partition refinement scheme. In Section 4, we introduce the key step of dynamic thresholding for the first sweep which quickly detects P-nodes in the recognition of Robinsonian matrices. Finally, we discuss in Section 5 progress made under lexicographic searches and sparse matrices.

2. Lexicographic Searches and Four Points Condition

We follow the standard graph notations with lower case for vertices and upper case for sets of vertices $V = \{v_1, \ldots, v_n\}$ or sets of edges

$E \subseteq V \times V$. The complete undirected graph with $E = V \times V$ is discarded as trivial for our purpose; similarly, only $E \neq \emptyset$ is relevant for our study. We denote $N(v)$ (resp. $N[v]$) the set of all vertices adjacent to vertex v, v excluded (resp. v included). Unlike most articles in graph theory where adjacency relationship depends on authors, we borrow the ternary operator from programming languages *if then else* conditional statement since it is very convenient wherever both the adjacent and not adjacent relationships are simultaneously invoked; $a?x : y$ means $(a, x) \in E$ together with $(a, y) \notin E$, for $E \subseteq G \times G$. By abuse of notation $a?x$ stands for adjacency and $a : y$ for non-adjacency between vertices; though the usual negation operator $/$ applies on adjacent (resp. not adjacent) relation $a?x$ (resp. $a : x$) since $a \not? x$ (resp. $a \not: x$) is equivalent to $a : x$ (resp. $a?x$), we prohibit its usage for ease of reading. The ternary operator helps avoiding long sentences to describe precise configurations, leaving the configuration easily retrieved from the formulas; for instance, an umbrella (frequently used in lexicographic searches) is an ordered triple $u < v < w$ such that $u?w : v$ with $v : w$. It is convenient to extend the neighborhood and ternary operators to subsets $X, Y \subseteq E$: $N(V) = \{u | \exists v \in v, u?v\}$ and adjacency $X?Y$ (resp. non-adjacency $X : Y$) means $x?y$ (resp. $x : y$) for all pairs of elements $x \in X$ and $y \in Y$. In order to concisely describe subsequent properties: x (resp. X) is said to be universal to a subset Y whenever $x?Y$ (resp. $X?Y$). Similarly, $M?N(M) : B$ is a very concise notation for a module, $C?C$ for a clique under self-loops, and $S : S$ for a stable set (ignoring self-loops). A claw is written on a stable set $\{b, c, d\}$ as

$$\begin{cases} a?\{b, c, d\} \\ \{b, c, d\} : \{b, c, d\} \end{cases} \qquad\qquad .$$

Using regular expressions from string pattern matching, the ternary notation allows to encompass path as $a?^{+}b$ (resp. $a :^{+} b$ for complement), so that an asteroidal triple (AT for short), i.e., a stable triple of points $\{a, b, c\}$ with paths connecting 2 points in the triple that avoid the neighborhood of the third, writes

$$\begin{cases} \{a, b, c\} : \{a, b, c\} \\ a?^{+}b : N(c) \\ a?^{+}c : N(b) \\ b?^{+}c : N(a) \end{cases}$$

where $a?^{+}b : N(c)$ denotes a path from a to b with length greater than 1 not adjacent to the neighborhood $N(c)$.

The ternary operator negation is blurring for subsets since $X \not? Y$ is clearly not equivalent to $X : Y$ (there may exist edges between X and Y in the former but not at all in the latter) and we prohibit this negation usage in the rest of this article for readability.

For an ordering σ, i-th element is denoted σ_i and consecutive elements from i to $j > i$ by $\sigma_{[i,j]}$. For $1 \leq i < j \leq n =\mid V \mid$, it is convenient to write $i <_\sigma j$ and avoid cumbersome notations $\sigma_i <_\sigma \sigma_j$ together with $\sigma^{-1}\sigma_i = i < j = \sigma^{-1}\sigma_j$ to mean element σ_i occurs before element σ_j in ordering σ and similarly for an interval $[i, j]$ of indices in σ.

For $i <_\sigma k$, the ordering σ is an

- I-ordering, iff $i?[i+1, k]$, i.e. $\sigma_i?\sigma_j$ for all $j \in [i+1, k]$
- UI-ordering, iff $i?[i+1, k]$ and $k?[i, k-1]$; indeed $i?k$ and $\{i?j, j?k\}$ for all $j \in [i+1, k-1]$.

2.1 Signed Four Points Condition

The *four points condition* on undirected graphs is written in concise notations, for all $j <_\sigma k <_\sigma l$ (BFS) and for all $i <_\sigma k <_\sigma l$ (DFS)

$$j?l : k \Rightarrow \exists i <_\sigma j, \quad i?k : l \qquad \text{(unsigned LexBFS)}$$
$$i?l : k \Rightarrow \exists i <_\sigma j <_\sigma k, \quad j?k : l \qquad \text{(unsigned LexDFS)}$$

For addressing the sparse matrix case, we have to switch the search between actual edges and missing edges in course of partition refinement versions of lexicographic searches; it is convenient to put such a binary attribute on edges and call it a sign to be distinguishable from usual weights put on edges. The pivoting rule in standard partition refinement algorithms (Habib et al., 2000) applies for undistinguishable edges but for our purpose will alternate between either edges with the convention that $-$ (resp $+$) edges are put in front (resp. at rear) of current partition refinement Figure 1 in accordance with natural ordering $-1, 0, +1$. In all figures, $i : j$ is drawn by a dashed edge between i and j and the sign attribute, if any, by the corresponding symbol close to the line $i?j$; graph vertex orderings mostly follow the alphabetical ordering or else are represented from left to right and the pivot is the leftmost vertex in the drawing context. The *signed* version of the four points condition writes

$$j? - l : k \Rightarrow \exists i <_\sigma j, \quad i? \pm k : l \qquad \text{(signed LexBFS)}$$
$$i? - l : k \Rightarrow \exists i <_\sigma j <_\sigma k, \quad j? \pm k : l \qquad \text{(signed LexDFS)}$$

The reader should pay attention to the difference with path notation where the (seemingly confusing) $+$ symbol operates on ternary adjacent/not adjacent splitting, while it operates on edge attribute in sparse recognition.

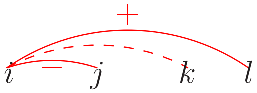

Figure 1. Signed pivoting along partition refinement: − ahead and + behind

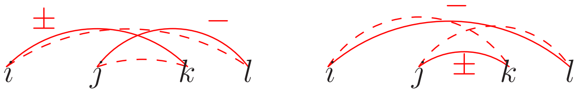

Figure 2. BFS-DFS four points signed condition

Definition 2.1 [minimal separator] A subset $S \subset V$ in a connected graph $G(V, E)$ is a separator if the induced subgraph $G(V \setminus S)$ is disconnected. It is minimal iff there exist at least 2 connected components C_1, C_2 whose neighborhood $N(V(C_1)), N(V(C_2))$ contains S; these are named full components.

Definition 2.2 [moplex] A moplex X in a graph G is a clique module whose neighborhood $N(X)$ is a minimal separator of the graph G. It is extended to a clique having an empty neighborhood.

From minimal separator condition, it implies that the clique module is maximal with respect to inclusion. With above notations, a moplex is defined as $(M?M, M?N(M) : C)$.

Unlike the unsigned cases (Berry and Bordat, 1998, Xu et al., 2013), the signed pivoting does not behave well with respect to moplex: let G be the ground set, σ the ordering produced on the relation $G \times G$ and X be the maximal clique module that the last element in σ belongs to; then, the moplex elements are not necessarily pivoted consecutively when (j, k) and (j, l) have same signs, see Figure 3. The definition encompasses directed version of partition refinement; however, to our knowledge, few is known on signed version of pivoting. We mention it for sake of self-containedness since we use it for P-nodes detection where the alternating front/rear insertion during partition refinement, brings a major achievement.

2.2 Unsigned Four Points Ending Moplex

Let X be the moplex containing the last vertex σ_n and $\text{CC}(X) = \{C_1, \ldots, C_p\}$ be the set of connected components defined by X where C_1 contains the first element σ_1, the structure of X, $CC(X)$ and $N(X) =$

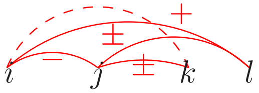

Figure 3. Signed moplex is not consecutively pivoted along partition refinement

$\cup_{c=1}^{p} N(C_c)$ depends on either breadth or depth first case. We note a decomposition in consecutive blocks of an ordering $\sigma = [B_1, \ldots, B_p]$ as $\sigma[1,p]$ since no confusion arises with subscripting of ordinals and similarly $\sigma[i,j]$ for consecutive blocks from i to $j > i$. By assumption of non-complete case, CC(X) exists and is not empty; furthermore $\sigma_1 : \sigma_n$.

2.2.1 Unsigned LexBFS Partition Refinement Case

Proposition 2.1 (Berry and Bordat, 1998) The moplex X containing σ_n is consecutive and the last connected component is a full component w.r.t $N(X)$.

$$\begin{cases} \sigma = \big[C_1 \cup N_1, \ldots, C_p \cup N_p, X\big] \\ N(C_1) = N_1 \\ N(C_p) = N(X) = \cup_{c=1}^{p} N_c \\ \sigma[c+1, p+1] ? N(C_c) : C_c, \quad \text{for all } c < p \quad \text{provided } \sigma[c] = C_c \cup N_c \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{starts with some element in } C_c \end{cases}$$

Proof.

- $\boxed{N_1 = N(C_1)}$ Clearly $N(\sigma_1) \subseteq N_1 \subseteq N(C_1)$; suppose $\sigma = \big[C_1^1, N_1^1, C_2, N_1^2, \ldots Y\big]$ then there exists $\sigma_j \in N_1^1$, $\sigma_k \in C_2$ and $\sigma_l \in N_1^2$ such that $j?k : l$ by pivoting rule; it contradicts σ_n being the last vertex since $N_1^1?(X \ni \sigma_n)$. Therefore $N(C_1) \subseteq N_1$ and $\sigma = \big[C_1 \cup N_1, \ldots, Y\big]$.

- $\boxed{X \text{ is consecutive}}$ If $\mid X \mid = 1$ there is nothing to prove. The case study:

  - $\boxed{\sigma = \big[\ldots, X^1, \ldots, C_c^1, \ldots, \sigma_n\big]}$ $\exists l < m < n$ with $\sigma_l, \sigma_n \in X$ and $\sigma_m \in C_c^1$; by moplex definition, $\sigma_m : \big\{\sigma_l, \sigma_n\big\} \subset X$ and therefore, by pivoting rule, σ_n could not be the last vertex.

  - $\boxed{\sigma = \big[\ldots, C_c^1, \ldots, X^1, \ldots, N_c^1, \ldots, \sigma_n\big]}$ Suppose C_c^1 is the earliest component part in σ and let $k < l < m < n$ with $\sigma_k \in C_c^1$, $\sigma_l \in X^1$ and $\sigma_m \in N_c^1$ for some connected component c such

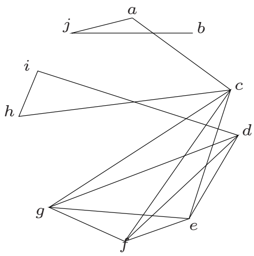

Figure 4. LexBFS moplex example

that $k?m : l$; therefore $\exists j < k$, $j?l : m$. Choose the smallest such j, since X is a moplex $j?n : m$ so that $\exists i < j$ with $i?m : n$. By moplex definition $i?m : l$ which implies $\exists h < i < j$ such that $h?l : m$ a contradiction to j being the smallest such index.

leads to $\sigma = \big[C_1 \cup N(C_1), \cup_{c=2}^{p}(C_c \cup N_c), X\big]$.

- $\boxed{\sigma[2, p+1]?N(C_1) : C_1}$ by induction on pivoting rule on $\sigma_{[1,n]} \setminus \cup_{c=1}^{r}(C_c \cup N_c)$, we get $\sigma = \big[C_1 \cup N(C_1), \ldots, C_p \cup N_p, X\big]$. There is nothing to prove for X, since $N(C_1)?X$ and $X : C_1$; once again, a case study proves the result:

  · $\boxed{\sigma = [C_1^1, N_1^1, \ldots, C_c^1, \ldots, X]}$ suppose $\exists j < k < l$ such that $\sigma_j \in N_1^1$, $\sigma_k \in C_c^1$ and $\sigma_l \in X$ such that $j?l : k$, then $\exists i < j$ such that $i?k : l$. If $\sigma_i \in C_1^1$ then 2 different connected components would be connected, a contradiction; otherwise,$\sigma_i \in N_1^1$ and $l?j : i$ contradicts X being a moplex. Notice σ starts with $\sigma_1 \in C_1$ so that the result holds for any sequence C_1^1, N_1^1, not only the earliest in σ; therefore, the results holds on $N(C_1) = \cup_r N_1^r$ and any part C_c^1.

  · $\boxed{\sigma = [C_1^1, N_1^1, \ldots, N_c^1, \ldots, X]}$ suppose $\exists j < k < l$ such that $\sigma_j \in N_1^1$, $\sigma_k \in N_c^1$ and $\sigma_l \in X$ such that $j?l : k$, then $\exists i < j$ such that $i?k : l$. If $\sigma_i \in C_1^1$ then by consecutiveness of $C_1 \cup N(C_1)$ we arrive at $c = 1$, a contradiction; otherwise,$\sigma_i \in N_1^1$ and $l?j : i$ contradicts X being a moplex (with the same remark as above, on sequence C_1^1, N_1^1).

Induction follows on induced subgraph on $\sigma_{[1,n]} \setminus (C_1 \cup N(C_1))$ provided each consecutive part $C_c \cup N_c$ start with an element in C_c. Then $N(C_p) = \cup_{c=1}^{p} N_p$ proves C_p being a full component different from X and $N(X)$ being a minimal separator.

■

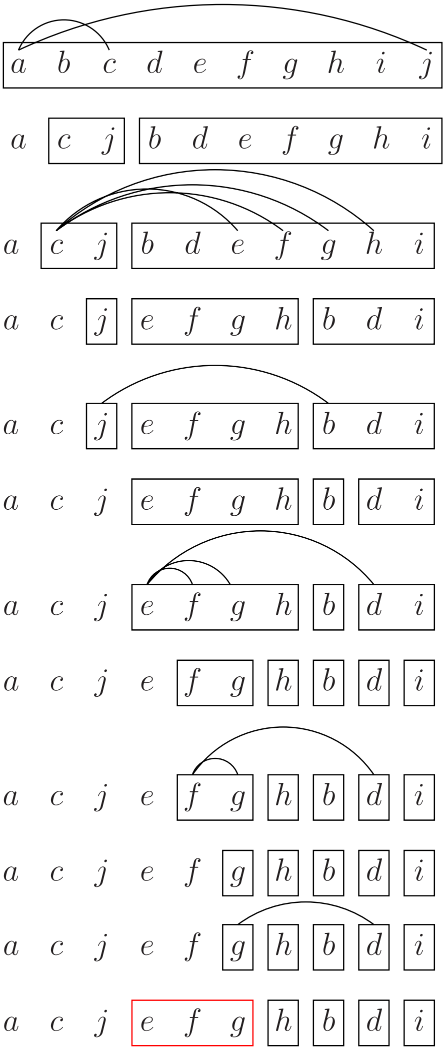

Figure 5. LexBFS standard pivoting (partition refinement)

In practical implementation, there is no way to distinguish an element of $N[X]$ from a connected component C_i, therefore the tie breaking rule has to be changed according to the specific separation goal.

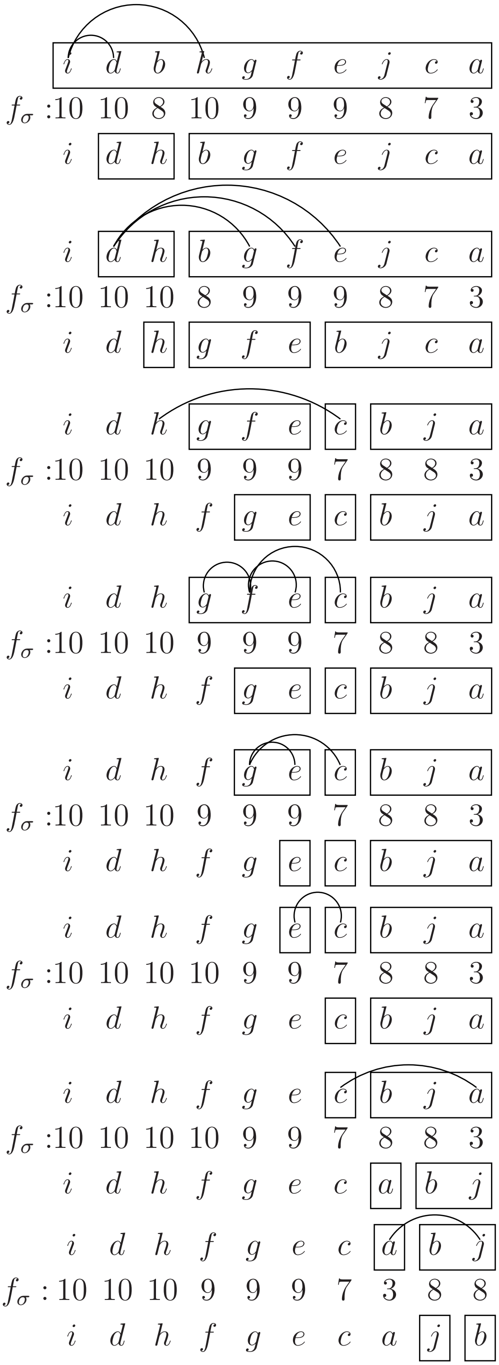

Figure 6. LexBFS Li-Wu pivoting (Li and Wu, 2014) along f_σ tie breaking (g, f, e) alternatives

2.2.2 Unsigned LexDFS Partition Refinement Case

Proposition 2.2 (Xu, Li, and Liang, 2013). *The moplex X containing σ_n is consecutive and the last connected component is a full component w.r.t $N(X)$ different from X.*

$$\begin{cases} \sigma = \big[C_1^1, N^1, (Y^2 \cup C_p^2), N^2, \dots, (Y^s \cup C_p^s), N^s, X\big] \\ N(C_p) = N(X) = \cup_{r=1}^{s} N^r \end{cases}$$

Proof. Let c_1^r (resp. c_{-1}^r) denotes the first (resp. last) element in $C_p^r \cup Y^r$ w.r.t. σ and correspondingly, let n_1^r (resp. n_{-1}^r) denote the first (resp. last) element in N^r for all $r \in [1, s]$.

- $\boxed{X \text{ is consecutive}}$ the case study

  · $\boxed{\sigma = \big[\dots, X^1, Y, C_c^1, \dots, X^2\big]}$ Let $i < k < l$ such that $\sigma_i \in X^1$, $\sigma_k \in C_c^1$ and $\sigma_l \in X^2$ then $i?l : k$ implies $\exists i < j < k$ such that $j?k : l$; by moplex definition and choosing the minimum k index, Y could not belong to $N(X)$, therefore $Y \subset C_c$ for some c.

  · $\boxed{\sigma = \big[\dots, X^1, Y, X^2\big]}$ from previous inclusion, $Y \subset (\cup C_c)$ for some parts of some connected components; by moplex definition, we have $X^1 ? X^2 : Y$ and $X^2 : Y$, then by pivoting rule $\sigma_n \in X^2$ could not be the last element in σ.

  leads to consecutiveness of the last moplex.

  Since there is nothing to prove for a single connected component, we assume $p > 1$ and $N(X) = \cup_{r=1}^{s} N^r$.

- $\boxed{\sigma = \big[Y^1 \cup C_p^1, N^1, \dots, (Y^r \cup C_p^r), N^r, \dots, (Y^s \cup C_p^s), N^s, X\big]}$ since $\sigma_1 \notin (C_p \cup X)$.

  Observe, it holds true

  · $\boxed{Y^r = C_c^r}$ for a single connected component c; let c be the connected component whose first element c_1^r of $C_p^r \cup Y^r$ belongs to, since $C_c : C_p$ for any $c \neq p$, by pivoting rule at c_1^r, which inserts $N(c_1^r)$ ahead all remaining elements $\sigma[2r, 2s+1]$, we have $Y^r = C_c^r$.

  · $\boxed{N^r \text{ and } C_c^r = Y^r \text{ are tree connected}}$ by definition of the pivoting rule, once more. In particular, $Y_1 \subseteq C_1 \ni \sigma_1$.

Then, consider the parts of any connected component C_c and parts N^r ahead, i.e. $r \in [1, t]$; the tree structure of C_c^{t+1} is refined w.r.t. N^r by

- $\boxed{\sigma_i ? c_1^{t+1} \text{ where } \sigma_i \neq n_{-1}^r \in N^r}$ let $\sigma_k = c_1^{t+1}$ and $\sigma_l \in X$ so that $i?l : k$, then $\exists i < j < k$ such that $j?k : l$, a contradiction since $\sigma_j \in N^r?X$ for all $r \in [1, t]$.

- $\boxed{n_{-1}^r : c_1^{r+1}}$ since at pivoting step n_{-1}^r, c_1^{r+1} is ahead of $N(n_{-1}^r)$ among elements in $\sigma[2r + 1, 2s + 1]$.

- $\boxed{c_{-1}^r : n_1^r}$ since at pivoting step c_{-1}^r, n_1^r is ahead of $N(c_{-1}^r)$ among elements in $\sigma[2r, 2s + 1]$.

- $\boxed{C_c^r \text{ parts are consecutive}}$ Suppose there exists 2 consecutive parts C_c^u and C_c^v separated by a part of a different component $C_{c'}^{u+2}$ with $v > u + 3$ then by connectedness there exists $\sigma_i \in C_c^u$, $\sigma_l = c_1^v \in C_c^v$ and $\sigma_k = c_1^{u+2} \in C_{c'}^{u+2}$ such that $i?l : k$ since $C_c : C_{c'}$, therefore $\exists i < j < k$ such that $j?k : l$. However, $\sigma_j \in N^{u+1}$ or else $c' = c$, so let $i' = j$, $k' = l$ and $l' > k'$ such that $\sigma_{l'}$ in moplex X then $i'?l' : k'$ and consequently $\exists i' < j' < k'$ such that $j'?k' : l'$; $\sigma_{j'} \notin N^r$ for $r \in (u, v)$ or else it contradicts $N^r?X$ by moplex definition, therefore $\sigma_{j'} \in C_{c''}$ for $c'' \neq c$ a contradiction too.

- $\boxed{n_{-1}^r ? c_1^{t+1} \text{ for all } r \in [1, t]}$ where C_c^{t+1} is the first part in σ of component C_c; suppose there exists $\sigma_i = n_{-1}^r \in N^r$, $\sigma_k = c_1^{t+1} \in C_c^{t+1}$ and $\sigma_l \in X$ such that $i?l : k$ then $\exists i < j < k$ such that $j?k : l$, then $\sigma_j \in C_{c'}^q$ for $r < q < t$ or else it contradicts $N^q?X$ by moplex definition; however $c' \neq c$ yields a contradiction too.

which leads to $N(C_c) = \cup_{r=1}^t (N^r)$ for all connected component $C_c = \cup_{r=q}^{t+1} C_c^r$ except possibly for the first and the last. As for the last component, if $N^s = \emptyset$ or $N^s \subset N(C_p^s)$ then we are done, otherwise there exists $\sigma_l \in N^s$ such that $\sigma_l : C_p$ then, by moplex definition, there exists $\sigma_i \in C_c^r$ for some $c \neq p$ such that $i?l : k$ for any $\sigma_k \in C_p$; but then N^s should be ahead of C_p by above arguments, whence $N^s = \emptyset$ for this case too; therefore, C_p is a full component different from X and $N(X)$ a minimal separator.

■

3. Four LexBFS Sweeps Interval Graphs Recognition

A graph is an interval graph if and only if it is chordal and AT-free, dates back to Lekkerkerker and Boland (1962/63). A characterization, independently observed by many authors (see Corneil, Olariu, and Stewart, 2009/10 for references) and more tightly related to LexBFS, relies on semi-umbrellas.

Theorem 3.1 [oriented semi-umbrellas] *A graph is an interval graph if and only if there exists a linear order on the vertices such that for all $u < v < w$, If $u?w$ then $u?v$.*

Unless a valid endpoint for an interval representation is known, a first LexBFS sweep is necessary to find such a *good* endpoint. In spite of the importance of the choice of a good endpoint, the first sweep is therefore a standard LexBFS sweep. Since the interval property is invariant under reversal, the following sweeps apply on reversal of the previous LexBFS ordering (an operative version of the underlying PQ-tree structure known as the Flipping lemma (Lekkerkerker and Boland, 1962/63).

The second sweep is a standard LexBFS sweep too, but collects the semi-umbrellas in a transitively reduced fashion, namely if $u < v_1 < \ldots < v_p$ then $u?\{v_i, v_{i+1}\}$ are collected as triples (u, v_i, v_{i+1}) in forward semi-umbrellas F; similarly if $v_1 < \ldots < v_q < u$ then triples (u, v_{i+1}, v_i) may be collected too, as backward semi-umbrellas B (for unit interval recognition); it is done online in $O(m)$. In Corneil et al. (2009/10), the triples are linked such that an item may be removed from all triples in O(the number of occurrences of the item) for easy deletion; the cardinalities are therefore stored for fast deletion. In our implementation, we use sparse matrix representation (in column major ordering indeed) so that a triple $(u, v, w) \mapsto T[v, u]$ and fulfills the required complexity. Of special importance is the largest neighbor in $N[u]$ along the LexBFS ordering for each u; obviously, it appears in a semi-umbrella, if any.

P. Li and Y. Wu (Li and Wu, 2014) observe that the original LexBFS algorithm (Corneil et al., 2009/10) simplifies to four sweeps by condensing two consecutive sweeps into the second as described above. They refer to the largest neighbor in a sweep σ as $f_\sigma(u)$; obviously $f_\sigma(u) \geq \sigma^{-1}(u)$ since $u \in N[u]$. Let us call the pivot part, the set of vertices S in the first part in the partition refinement; equivalently, the set of vertices with the lexicographically largest label in the labeling scheme. The standard LexBFS breaks ties among all elements in S by selecting the first, whence the doubly linked list efficiency. Let σ be the LexBFS ordering obtained by the second sweep; then, they apply a third LexBFS sweep ρ starting with $\sigma(n)$ as first pivot, with a special tie breaking rule at each pivot part S. let α (resp. β) be the elements in S appearing first (resp. last) in σ, let w be any vertex in S with maximum $f_\sigma(w)$, then select the pivot as:

- α if it is not finished, i.e. forward semi-umbrellas remain $F_\sigma(\alpha) \setminus F_\rho(\alpha) \neq \emptyset$
- β if $f_\sigma(w) \leq \beta$, any semi-umbrella w.r.t. w occurs before β
- w otherwise.

It may be confusing to refer to forward umbrellas without stressing the sweep which they are collected in: for **I**-ordering, we have to check consecutiveness of forward umbrellas, hence the relevance of forward umbrellas for the second and fourth sweeps, erasing the apparent ambiguity. The rule is fine but expensive since it requires a three parallel LexBFS sweep after sorting w.r.t. f_σ the largest neighbor obtained in second sweep. Their proof implicitly involves the largest neighbor in the topological sort of forward semi-umbrellas as a necessary condition in Theorem 3.1; then, an extra sweep (on reversed ordering) folds ordered semi-umbrellas into consecutive unless it is not an interval graph.

First, we claim that the parallel sweep further simplifies to 2 parallel sweeps, one on the decreasing order of largest neighbor in second sweep and the other on the reversal of second sweep, respectively ρ_3 and ρ_2 in (Li and Wu, 2014) notations with pivot w and β respectively.

Proposition 3.1 [reversal invariance of ρ_2 parts] *The refined parts in ρ_2 capture refined parts in both ρ_2 and ρ_1*

Proof. Let S be a pivot part in the two parallel LexBFS sweep; if S is a clique there is no umbrella possibility within S whatever pivot is chosen. Tie breaking rule simplifies to a three way branch, either endpoint of the pivot part of ρ_2 or the pivot of ρ_3 (equivalently an item with largest neighbor in pivot part which may differ from largest neighbor associated with either endpoint of ρ_2). Assuming parts in ρ_2 remain invariant under reversal, i.e. reversal of $\rho_2 = \rho_1$, whatever the pivot is among w, β and α for implicit ρ_1 since LexBFS refinement keeps initial ordering within parts, the parts in ρ_2 remain invariant by reversal.
■

Since the pivot part is the same up to reversal for ρ_2 and ρ_1 in parallel sweeps, the doubly linked list structure for items is enhanced by circular shift to retrieve either endpoint. Instead of two $O(m+n)$ parallel sweeps, a question is raised about the overall complexity of sorting the pivot part w.r.t. the largest neighbor on demand (instead of the offline counting sort after the second sweep and a 2 parallel sweeps). According to Corneil et al. (2009/10), let's say α flies whenever its largest neighbor belongs to a refined part different from the pivot part. If the pivot part is singleton or the classical pivot (head of the pivot part) α flies, the pivot should be α and there is no need for w flying test in Li-Wu sense.

Finally, using the semi-umbrellas structure, SU (Corneil et al., 2009/10), which keeps track of the remaining forward semi-umbrellas at each item, along with a reference to the largest neighbor in second sweep, the third sweep operates (on the reversed second sweep ordering) with the following rule:

Definition 3.1 [interval graph tie breaking]

- if $\mid SU[\alpha] \mid > 0$ or α flies, then pivot=α (head of pivot part)
- otherwise, for each w in pivot part, sort w.r.t. unassigned largest neighbor flying in $O(|\text{pivot part}|)$; if any then pivot=w else pivot=β (tail of pivot part)

and the proof of correctness remains unchanged. The complexity of maintaining the semi-umbrellas structure, delete (v) and $\mid SU[v] \mid$ remains $O(\deg(v))$ (Corneil et al., 2009/10), therefore the amortized (sorting on demand) complexity is $O(m+n)$, since the worst case of sorting on demand is $O(n^2)$ for the complete graph, i.e. $O(m)$. It improves over parallel sweeps in $O(m+n)$ by significantly scaling down the constant. We report examples from Li and Wu (2014) under various starting permutations. The four sweeps are followed by a realizer, i.e. an ordering among the set of permutations fulfilling the interval property, giving the endpoints of the interval associated with its corresponding item in last permutation.

3.1 Corneil's Example

As a teaching example for LexBFS structure description (Corneil et al., 2009/10), it is cited in the four sweeps algorithm too (Li and Wu, 2014).

$$\begin{cases} \begin{bmatrix} \delta \\ \sigma \\ \rho' \\ \tau' \end{bmatrix} = \begin{bmatrix} 20 & 8 & 2 & 4 & 21 & 16 & 15 & 9 & 12 & 13 & 11 & 14 & 17 & 10 & 18 & 7 & 19 & 6 & 5 & 3 & 1 & 22 \\ 22 & 4 & 3 & 2 & 5 & 6 & 7 & 8 & 9 & 18 & 17 & 12 & 14 & 13 & 11 & 15 & 16 & 10 & 19 & 20 & 21 & 1 \\ 1 & 2 & 20 & 8 & 4 & 6 & 7 & 9 & 18 & 17 & 12 & 11 & 13 & 15 & 14 & 16 & 10 & 19 & 5 & 3 & 21 & 22 \\ 22 & 4 & 21 & 20 & 8 & 2 & 19 & 18 & 17 & 9 & 12 & 16 & 15 & 13 & 14 & 11 & 10 & 7 & 6 & 5 & 3 & 1 \end{bmatrix} \\ \quad \mathbf{I} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 \\ 2 & 21 & 4 & 6 & 19 & 22 & 8 & 10 & 11 & 18 & 16 & 13 & 14 & 16 & 15 & 17 & 17 & 19 & 20 & 20 & 21 & 22 \end{bmatrix} \end{cases}$$

$$\begin{cases} \begin{bmatrix} \delta \\ \sigma \\ \rho' \\ \tau' \end{bmatrix} = \begin{bmatrix} 22 & 4 & 3 & 2 & 5 & 6 & 7 & 8 & 9 & 18 & 17 & 12 & 14 & 13 & 11 & 15 & 16 & 10 & 19 & 20 & 21 & 1 \\ 1 & 2 & 20 & 8 & 4 & 19 & 18 & 17 & 9 & 12 & 16 & 15 & 13 & 11 & 14 & 10 & 7 & 6 & 5 & 3 & 21 & 22 \\ 22 & 4 & 2 & 20 & 8 & 6 & 7 & 9 & 18 & 17 & 12 & 11 & 13 & 15 & 14 & 16 & 10 & 19 & 5 & 3 & 21 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 13 & 12 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 \end{bmatrix} \\ \quad \mathbf{I} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 \\ 2 & 20 & 4 & 22 & 6 & 8 & 9 & 20 & 18 & 11 & 13 & 15 & 17 & 14 & 16 & 16 & 18 & 19 & 19 & 21 & 21 & 22 \end{bmatrix} \end{cases}$$

$$\left\{\begin{array}{l}\begin{bmatrix}\delta\\ \sigma\\ \rho'\\ \tau'\end{bmatrix} = \begin{bmatrix} 4 & 14 & 12 & 8 & 9 & 13 & 2 & 11 & 15 & 16 & 17 & 10 & 18 & 7 & 19 & 6 & 20 & 5 & 3 & 21 & 22 & 1\\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 18 & 17 & 12 & 16 & 15 & 13 & 11 & 14 & 10 & 19 & 20 & 21 & 22\\ 22 & 4 & 2 & 20 & 8 & 6 & 7 & 9 & 18 & 17 & 12 & 11 & 13 & 15 & 14 & 16 & 10 & 19 & 5 & 3 & 21 & 1\\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 13 & 12 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22\end{bmatrix}\\ \qquad\mathbf{I} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22\\ 2 & 20 & 4 & 22 & 6 & 8 & 9 & 20 & 18 & 11 & 13 & 15 & 17 & 14 & 16 & 16 & 18 & 19 & 19 & 21 & 21 & 22\end{bmatrix}\end{array}\right.$$

3.2 Ma's Rxample

This famous example was personally given by T. Ma to D.G. Corneil as a counterexample to his early version of multisweeps algorithm for interval graph recognition; as such, it is another teaching example to probe the various multiple LexBFS sweeps algorithms (Li and Wu, 2014).

$$\left\{\begin{array}{l}\begin{bmatrix}\delta\\ \sigma\\ \rho'\\ \tau'\end{bmatrix} = \begin{bmatrix} 1 & 2 & 15 & 5 & 14 & 6 & 9 & 8 & 10 & 11 & 12 & 13 & 7 & 4 & 16 & 3 & 17\\ 17 & 16 & 5 & 2 & 4 & 6 & 7 & 8 & 10 & 9 & 12 & 11 & 13 & 14 & 15 & 3 & 1\\ 1 & 2 & 16 & 5 & 4 & 6 & 14 & 9 & 8 & 10 & 12 & 11 & 13 & 7 & 15 & 3 & 17\\ 17 & 16 & 5 & 2 & 15 & 14 & 9 & 6 & 13 & 12 & 11 & 10 & 8 & 7 & 4 & 3 & 1\end{bmatrix}\\ \qquad\mathbf{I} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17\\ 2 & 4 & 15 & 17 & 6 & 8 & 13 & 15 & 10 & 12 & 12 & 13 & 14 & 14 & 16 & 16 & 17\end{bmatrix}\end{array}\right.$$

$$\left\{\begin{array}{l}\begin{bmatrix}\delta\\ \sigma\\ \rho'\\ \tau'\end{bmatrix} = \begin{bmatrix} 17 & 16 & 5 & 2 & 4 & 6 & 7 & 8 & 10 & 9 & 12 & 11 & 13 & 14 & 15 & 3 & 1\\ 1 & 2 & 3 & 4 & 6 & 5 & 14 & 9 & 13 & 12 & 11 & 10 & 8 & 7 & 15 & 16 & 17\\ 17 & 16 & 2 & 5 & 4 & 6 & 14 & 9 & 8 & 10 & 12 & 11 & 13 & 7 & 15 & 3 & 1\\ 1 & 2 & 3 & 4 & 6 & 5 & 7 & 8 & 10 & 9 & 11 & 12 & 13 & 14 & 15 & 16 & 17\end{bmatrix}\\ \qquad\mathbf{I} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17\\ 2 & 16 & 4 & 6 & 14 & 16 & 8 & 10 & 12 & 14 & 12 & 13 & 13 & 15 & 15 & 17 & 17\end{bmatrix}\end{array}\right.$$

$$\left\{\begin{array}{l}\begin{bmatrix}\delta\\ \sigma\\ \rho'\\ \tau'\end{bmatrix} = \begin{bmatrix} 14 & 2 & 5 & 15 & 9 & 6 & 12 & 11 & 10 & 13 & 8 & 7 & 4 & 16 & 3 & 1 & 17\\ 17 & 16 & 5 & 2 & 4 & 6 & 7 & 8 & 10 & 9 & 11 & 12 & 13 & 14 & 15 & 3 & 1\\ 1 & 2 & 16 & 5 & 4 & 6 & 14 & 9 & 8 & 10 & 12 & 11 & 13 & 7 & 15 & 3 & 17\\ 17 & 16 & 5 & 2 & 15 & 14 & 9 & 6 & 13 & 12 & 11 & 10 & 8 & 7 & 4 & 3 & 1\end{bmatrix}\\ \qquad\mathbf{I} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17\\ 2 & 4 & 15 & 17 & 6 & 8 & 13 & 15 & 10 & 12 & 12 & 13 & 14 & 14 & 16 & 16 & 17\end{bmatrix}\end{array}\right.$$

4. Dynamic Thresholding

4.1 Extreme Rays of the Cone of Robinsonian Matrices

As a special case of Monge matrices (Rudolf and Woeginger, 1995; Klinz, Rudolf, and Woeginger, 1995), it has long been known that Robinsonian matrices form a cone whose extreme rays have a simple structure (P. Arabie and L. Hubert (Hubert and Arabie, 1994) already mention it, at least implicitly). An upper trapezoidal staircase matrix is defined as a binary(0-1) matrix such that the sequence of rightmost 0 index $r_0(i)$ in row i is non-decreasing along the rows $r_0(i) \leq r_0(i+1)\ldots$; correspondingly, we will note $l_0(i)$ the leftmost 0 in row i to account for lower trapezoidal staircase matrix.

Up to symmetry and due to the 0-diagonal structure of Robinsonian dissimilarities, the following staircase Robinsonian dissimilarities form the extremal rays of the cone of the $n \times n$ Robinsonian dissimilarities.

$$U^{ij}_{pq} = 1 \text{ for all } 1 \leq p \leq i, r_0(p) \leq q \leq n$$
$$L^{ij}_{pq} = 1 \text{ for all } i \leq p \leq n, 1 \leq q \leq l_0(p)$$

Let $\mathcal{U} = \{(U^{ij}, r_0^{ij}) | 1 \leq i < n, i+1 \leq j < n\}$, $\mathcal{L} = \{(L^{ij}, l_0^{ij}) | j+1 \leq i < n, 1 \leq j < n\}$ where r_0^{ij} (l_0^{ij} resp.) is a non-decreasing sequence of indices of rightmost (leftmost resp.) 0 associated with U^{ij} (L^{ij} resp.) rows.

Lemma 4.1 *Every non-negative Robinsonian dissimilarity* $\mathbf{D}$ *is an affine combination of* $\{\mathcal{U} \cup \mathcal{L}\}$

$$\mathbf{D} = \sum \upsilon_{ij} U^{ij} + \sum \lambda_{ij} L^{ij}, \quad \upsilon_{ij} \geq 0, \lambda_{ij} \geq 0$$

Proof. Suppose the contrary, i.e. there exist non-negative Robinsonian dissimilarities which cannot be represented as an affine combination of matrices in $\{\mathcal{U} \cup \mathcal{L}\}$. Let $\mathbf{D}$, w.l.o.g. an upper trapezoidal, non-negative Robinsonian dissimilarity be a counterexample with the maximum number of 0 entries; let r_0^D be the sequence of indices of rightmost 0 along the rows. Then, find the minimum entry, say $\alpha = d_{k(r_0^D(k)+1)}$ in $\mathbf{D}$ adjacent to each rightmost 0, taking the lowest rightmost in case of ties. Next, from r_0^D, select in $\mathcal{U}$ the associated non-negative $(U^{ij}, r_0^{ij} = r_0^D)$. Clearly $\mathbf{D}' = \mathbf{D} - \alpha(U^{ij})$ is still a Robinsonian dissimilarity with at least one more zero $d'_{k,(r_0^D(k)+1)}$ contradicting the assumption.
■

On the contrary, anti-Robinsonian matrices do not form a cone as the counterexample $\mathbf{A} + \mathbf{B} = \mathbf{C}$, shows for $i < j < k$

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 4 \\ 3 & 0 & 3 \\ 4 & 3 & 0 \end{bmatrix}$$
$$\max(a_{ij}, a_{jk}) \geq a_{ik}, \quad \max(b_{ij}, b_{jk}) \geq b_{ik}$$
$$\max(c_{ij}, c_{jk}) < c_{ik} = a_{ik} + b_{ik}$$

The LexBFS partition refinement algorithm splits neighbors of the pivot in a part just in front of the parts they are extracted from, hence the doubly linked list structure within part, for easy extraction of neighbors, as well as among parts, for easy insertion of new parts. It is well known that LexBFS operates on the graph complement provided insertion takes place at rear. This observation is crucial in Robinsonian matrices case since both dissimilarity and similarity are in complement correspondence $s_{ij} = M - d_{ij}$ where M is the maximum value in $\mathbf{D}$ (under non-negative entries mild assumption). As a consequence, alternating front/rear insertion in first LexBFS sweep still yields a valid endpoint for an underlying interval graph recognition. A preprocessing counts the number of times the maximum value occurs per row then sorts the rows appropriately by counting sort in overall $O(n^2)$.

Finally, observe that among extreme rays of the cone those that *touch* the diagonal divide the dissimilarity in independent components; in other words, upper right corner belonging to (up to scaling) $\mathcal{U}$ yields a P-node in the PQ-tree. Dividing a matrix in such P-node requires $O(n^2)$. We recall, from Section 1 that difficult recognition instances come from such multiple common values, therefore it is likely to occur in Robinsonian matrices. The quite effective trick consists in alternating front/rear insertion at each pivot step along dynamic thresholding: let m, M be respectively the minimum and the maximum over the unassigned items for the pivot p then we insert all unassigned v such that $d_{pv} \leq m$ (resp. $d_{pv} \geq M$) in front (resp. at rear) of the part it is extracted from, in two consecutive passes for the same pivot. The complexity still remains $O(m) = O(n^2)$ for the whole sweep. The lemma on the extreme rays structure guarantees that we get a P-node, together with a valid endpoint for interval representation at the end of the first sweep.

Furthermore, within the same complexity $O(n^2)$, it is clear that we can retrieve a threshold for each diagonal block where a clash w.r.t. the Robinson property occurs in the upper left corner. Therefore, we can conquer each diagonal block with a different threshold, a feature not apparent in previous algorithms (both PQ-tree based and LexBFS based).

4.2 The Partition Avoiding Lemma

After B. Dietrich (Dietrich, 1990), let us define

Definition 4.1 [Robinson violated umbrella] An ordered sequence $i < j < k$ is violated for dissimilarity $\mathbf{D}$, if $\max(d_{ij}, d_{jk}) > d_{ik}$; it means d_{ik} is defined and at least one among $d_{ij} \leq d_{ik}$ and $d_{jk} \leq d_{ik}$ inequalities is violated.

To each non-violated umbrella $i < j < k$, we associate a semi-umbrella $i?\{j, k\}$ in the (interval) graph sense. As a consequence, a sparse dissimilarity is Robinsonian if and only if the non-violated semi- umbrellas fulfill the **UI** property.

To avoid confusion with above notation for a path in a graph, let us denote a sequence $i <_\sigma j <_\sigma k$... w.r.t. a partitioning order σ, as an ordered tuple $(i, j, k...)_\sigma$ and a Robinson violated umbrella as $\overline{d(i, j, k)_\sigma}$ for short; by a little abuse of notation, $d(i, j, k)_\sigma$ means the semi-umbrella is not violated (d_{ik} is not defined or else if d_{ij} is defined then $d_{ij} \leq d_{ik}$ and if d_{jk} is defined then $d_{jk} \leq d_{ik}$).

Definition 4.2 A pivot x in the partition refinement splits y, z belonging to the same part (denoted by $(x \mid\ y, z)_\sigma$) if x occurs first in σ partitioning order, $d_{xy} < d_{xz}$ for both defined entries and for all $u <_\sigma x$, $d(u, y, z)_\sigma$

Again by abuse of notation, $(x \nmid\ y, z)_\sigma$ means that x does not split $(y, z)_\sigma$ independently of its occurence in σ (not necessary the first). A Robinson ordering π for a dissimilarity $\mathbf{D}$ means that the symmetrically permuted matrix $\mathbf{D}_\pi$ is Robinsonian. By a slight abuse of language, a dissimilarity is still said Robinsonian whenever such a Robinson ordering exists.

Definition 4.3 Given distinct elements x, y, z, a path from x to z avoiding y is a sequence $(x = u_0, u_1, \ldots, u_k, u_{k+1} = z)$ such that $\overline{d(u_i, y, u_{i+1})}$ for all $0 \leq i \leq k$.

Once more, we need a slight modification to handle missing entries, since $\overline{d(u_i, y, u_{i+1})}_\sigma$ required $d_{u_i u_{i+1}}$ to be defined by Definition 4.1.

Definition 4.4 Given distinct elements x, y, z, a sequence from x to z avoiding y is a sequence $(x = u_0, u_1, \ldots, u_k, u_{k+1} = z)$ such that there exists a subsequence which is a path from x to z avoiding y.

As an easy consequence of this definition,

Lemma 4.2 *Let a dissimilarity be sparse Robinsonian; if there exists a sequence from x to z avoiding y, then y cannot lie between x and z in any Robinson ordering π.*

We borrow the refinement rule in Laurent and Seminaroti (2016) that sorts the *neighbors* ys of the pivot x within the same part w.r.t. increasing values of defined d_{xy} and then adapt their proofs to handle missing entries d_{xz} in the same part. In particular, we need extending the splitting from a pair to a part.

Definition 4.5 A pivot w in the partition refinement splits a part p (denoted by $(w \mid\ p)_\sigma$) if $(w \mid\ y, z)_\sigma$ for some $y, z \in p$.

To emphasize the separation, the part p is denoted by $[y, z]$ for ease of reading but may contain more elements.

Lemma 4.3 [Laurent and Seminaroti, 2016] *Let $(x, y, z)_\sigma$ in a partition refinement of a P-node and suppose there exists an ordering π such that $\mathbf{D}_\pi$ is Robinsonian with $(x, z, y)_\pi$, then for all $u \leq_\sigma x$ such that d_{uz} is defined, $(u \not| [y, z])_\sigma$.*

Proof. W.l.o.g. we assume d_{xy} being defined. Suppose $(x \mid y, z)_\sigma$ then $d_{xy} < d_{xz}$ and $d(x, z, y)_\pi$ yields the contradiction $d_{xz} \leq d_{xy} < d_{xz}$ on both defined values; therefore $(x \not| y, z)_\sigma$ starts the induction. By contradiction, let us assume there exists a pivot $w <_\sigma x$ such that $(w \mid y, z)_\sigma$, then $d_{wy} < d_{wz}$ and for all $u <_\sigma w$, $d(u, y, z)_\sigma$; observe that both d_{wy} and d_{wz} are defined and suppose $(w, z, y)_\pi$ for some Robinson ordering π, then we get the contradiction $d_{wz} \leq d_{wy} < d_{wz}$; whence,

$$\begin{cases} (w, x, y, z)_\sigma \\ (w \mid [y, z])_\sigma, \quad d_{wy} < \max(d_{wx}, d_{wz}) \\ d(x, z, w)_\pi \end{cases}$$

where the part stresses the actual separation and the inequality holds trivially if d_{wx} is missing, else it reflects the splitting of part w.r.t the 2 separated values $d_{wy} < d_{wz}$.

We claim that there exists $v \leq_\sigma w$ such that $(v \mid x, y)_\sigma$ or else, x belongs to the same part as $[y, z]$ and since w is the first occurrence that splits it, the refinement at pivot w yields

- if d_{xw} is missing: x is assigned to the part associated with y at pivot w or else it contradicts $(w, x, y, z)_\sigma$.
- if d_{xw} is defined: $d_{wy} < d_{wz} \leq d_{xw}$ by the last 2 relations, i.e. $(w, y, x)_\sigma$, a contradiction to the first relation.

Since $(w \mid [y, z])_\sigma$, it implies $(v \mid [x, z])_\sigma$ in part sense.

Observe $d(x, v, y)_\pi$ or else $d(x, y, v)_\pi$ and $(v \mid x, y)_\sigma$ yields the contradiction $d_{xv} < d_{yv} \leq d_{xv}$. We claim further, $v <_\pi z$ if d_{zv} is defined, or else suppose $d(x, z, v)_\pi$ for contradiction, then $d_{zv} \leq d_{xv} < d_{vy}$ since $(v \mid x, y)_\sigma$; it implies $\{x, z\}$ are in the same part after pivoting v, i.e. $(v \mid [y, z])_\sigma$, a contradiction to w being the first splitting, $(w \mid [y, z])_\sigma$.

Altogether, it achieves the induction step with the reversal $\tilde{\pi}$ of π, which is a Robinson ordering too,

$$\begin{cases} (v, w, x, z)_\sigma \\ (v \mid [x, z])_\sigma, \quad d_{vx} < \max(d_{vw}, d_{vz}) \\ d(w, z, v)_{\tilde{\pi}} \end{cases}$$

which concludes the proof.
■

Notice, on the contrary to dense case, the proof is not constructive under missing entries. Consider the induction step and assume d_{vz} is missing; let examine the conditions for $z <_\pi v$ contradiction to hold: suppose $d(x, z, v, y)_\pi$, since $d(v, x, y, z)_\sigma$ and d_{xy} is defined, we have $d_{xy} \leq d_{vy} \leq d_{xy}$ from respectively σ and π orderings, i.e. $d_{xy} = d_{vy}$; moreover, from $d(v, w, y)_\sigma$, it holds $d_{wy} \leq d_{vy}$. As a consequence, if d_{xz} is defined (as in dense case), then $d_{xz} \leq^\pi d_{vx} <^\sigma d_{vy} = d_{xy} = d_{xz}$ since $(x \not\mid y, z)_\sigma$ for last equality, where superscripts stress the ordering used to derive the comparison of entries (no confusion may arise with comparisons of indices in the orderings); clearly it is a contradiction, and put further shed on the role of defined entries d_{vz} and d_{xz}. However, if d_{vz} is missing but d_{xw} is defined then $d(x, z, w, v, y)_\pi$ leads to the chain rule: $d_{xw} \leq^\pi d_{xv} <^\sigma d_{vy} = d_{wy} <^\sigma d_{wz} \leq^\pi d_{xw}$, by using $(v \mid x, y)_\sigma$ and $(w \mid y, z)_\sigma$, a contradiction; it proves the claim $v <_\pi z$ and the induction step under the weak sense of $d_{vx} < \max(d_{vw}, d_{vz})$.

Corollary 4.1 *Let* $\mathbf{D}$ *be a sparse Robinsonian dissimilarity,* σ *an ordering obtained by the sorting refinement rule on a* P*-node, and consider distinct elements* $(x, y, z)_\sigma$ *then*

$$\begin{cases} d_{xy} \leq \max(d_{xz}, d_{yz}) \\ \textit{if } d(x, z, y)_\pi \textit{ for some } \pi \textit{ then the sequence } (x, z) \textit{ does not avoid } y \end{cases}$$

Proof. First, assume by contradiction $d_{xy} > \max(d_{xz}, d_{yz})$ and let $x <_\pi y$ in $\mathbf{D}_\pi$ Robinsonian dissimilarity; notice that d_{xy} is defined and at least one of d_{xz} and d_{yz} is appropriately defined. $d(x, y, z)_\pi$ (resp. $d(z, x, y)_\pi$) implies $d_{xy}, d_{yz} \leq d_{xz} = \max(d_{xz}, d_{yz})$ (resp. $d_{zx}, d_{xy} \leq d_{zy} = \max(d_{zx}, d_{zy})$) a contradiction. Therefore, $d(x, y, z)_\pi$ and by Lemma 4.3 we have for all $u \leq_\sigma x$ such that d_{uz} is defined, $(u \not\mid y, z)_\sigma$; for $u = x$ it specializes to $d_{xz} \leq d_{xy}$ contradicting the assumption. Last, if the sequence (x, z) avoids y then by the lemma, $d_{xz} > d_{xy}$ a contradiction to $d(x, z, y)_\pi$.
■

Lemma 4.4 [Partition Avoiding Lemma] *Let* $(x, y, z)_\sigma$ *in a partition refinement of a* P*-node of a sparse Robinsonian dissimilarity. If* $d(x, z, y)_\pi$ *for some Robinson ordering* π *whose reversal* $\tilde{\pi}$ *is a Robinson ordering too, then there does not exist a sequence* $(x = u_0, u_1, \ldots, u_k, u_{k+1} = z)$ *avoiding* y *with* $u_i <_\sigma z$*, for all* $1 \leq i \leq k$*.*

Proof. Since a sequence contains a path as a subsequence, the result follows by applying the path avoiding lemma (Laurent and Seminaroti, 2016).
■

However, unlike in dense case, the proofs are not constructive; in particular, nothing is said about assigning a missing entry to a part along the partition refinement when d_{vz} is missing as observed above and we state

Conjecture 4.1 There exists a *sorting* rule that assigns missing entries to parts such that the lemma Lemma 4.3 holds under missing d_{vz}.

Notice the importance of operating on a P-node in the proofs, since the missing entries deeply affects the partition refinement, so the decomposition using the dynamic thresholding rule on sparse submatrices (whose rows have the same maximum value) strongly reduces ties on subsequent sweeps using the *sorting* refinement rule. Up To this restriction to P-node, the multisweep algorithm using the sorting refinement rule in Laurent and Seminaroti (2016) applies provided a good starting point is found; it leads to the $O(n^2 + nk\log(n))$ complexity where k is less than the number of different entries as stated in this reference.

Last, the **UI** property of non-violated semi-umbrellas suggests, after the interval graph recognition case, that the number of sweeps should not exceed 4 and therefore, achieve an optimal recognition algorithm for sparse Robinsonian matrices as well. Apart from assignment of missing entries to parts, a challenging task remains to deal with missing entries in the Li-Wu framework (Li and Wu, 2014), to guarantee that the parts in the 3 parallel orderings sweep, contain the same indices, on the one hand (or else by the **UI** property, it could not be sparse Robinsonian); even though this condition is likely satisfied, it does not guarantee that the endpoints of the pivoting parts do not involve missing entries, on the other hand.

5. Discussion

5.1 Practical Issues

It is customary to account for $O(m+n)$ complexity, m the number of edges and n the number of vertices, for a LexBFS sweep; it assumes in a multiple sweeps algorithm, like interval graph recognition, that the mapping of vertices changes from one sweep to the other. Dynamic renumbering of vertices happens to become quite cumbersome for huge graphs that overflow in memory and even intractable if the neighborhood is given by a read-only media. For a LexDFS sweep, the vertices extracted from each part have to keep their relative ordering in the newly created part ahead of the current refinement; it introduces a $O(\log m)$ factor for the worst case complexity. In our implementation, the mapping is done online, in the same way for both LexBFS and LexDFS ; it requires sorting the vertices extracted from a part, according to the permutation entering the sweep, unless the cardinality of the extracted vertices is 1 or equal to the whole part. Our belief is that the amortized complexity for LexBFS case is far below the $O(\log m)$ worst case, following the observations about all different/same values for Robinsonian matrices (the occurence of a singleton or the whole part adjacent to the pivot is not scarce at all). If a dedicated study proves the amortized complexity far above $O(m)$, then it raises the question whether the doubly linked list structure is well suited.

5.2 Sparse Example

5.2.1 Dynamic Thresholding Rule

Starting from a sparse matrix drawn from a dense Robinsonian dissimilarity (Préa and Fortin, 2014),

$$\begin{bmatrix}
0 & 9 & 2 & 11 & 6 & 11 & 6 & & 9 & 11 & 6 & 11 & 6 & & 11 & 11 & 9 & 11 & 6 \\
9 & 0 & 9 & 11 & 2 & 11 & & 6 & 1 & 11 & 6 & & 6 & & 11 & & 1 & 11 & 3 \\
2 & 9 & 0 & 11 & 6 & 11 & 6 & 6 & 9 & 11 & 6 & 11 & 6 & & 11 & 11 & & 11 & 6 \\
11 & 11 & 11 & 0 & 11 & 8 & 11 & 11 & 11 & & 11 & 8 & & 11 & 1 & 8 & & 2 & \\
6 & 2 & 6 & 11 & 0 & 11 & & 4 & 2 & 11 & 4 & & & 6 & 11 & 11 & 2 & 11 & \\
11 & 11 & 11 & 8 & 11 & 0 & & 11 & 11 & 1 & 11 & 5 & 11 & & 6 & 3 & 11 & 6 & 11 \\
6 & & 6 & 11 & & & 0 & 4 & 3 & 11 & 4 & 11 & 4 & & 11 & 11 & 3 & 11 & 2 \\
 & 6 & 6 & 11 & 4 & 11 & 4 & 0 & 5 & 11 & 1 & 11 & & 4 & & 11 & 5 & 11 & 4 \\
9 & 1 & 9 & 11 & 2 & 11 & 3 & 5 & 0 & 11 & 5 & 11 & 6 & 9 & 11 & 11 & & & 3 \\
11 & 11 & 11 & & 11 & 1 & 11 & 11 & 11 & 0 & 11 & 5 & & 11 & 7 & 2 & 11 & 6 & 11 \\
6 & 6 & 6 & 11 & 4 & 11 & 4 & 1 & 5 & 11 & 0 & 11 & 2 & 4 & 11 & 11 & 5 & 11 & 4 \\
11 & & 11 & 8 & & 5 & 11 & 11 & 11 & 5 & 11 & 0 & 11 & 11 & 2 & 5 & 11 & 1 & 11 \\
6 & 6 & 6 & & & 11 & 4 & & 6 & & 2 & 11 & 0 & 4 & 11 & 11 & 6 & 11 & \\
 & & & 11 & 6 & & & 4 & 9 & 11 & 4 & 11 & 4 & 0 & 11 & 11 & & 11 & 6 \\
11 & 11 & 11 & 1 & 11 & 6 & 11 & & 11 & 7 & 11 & 2 & 11 & 11 & 0 & 7 & & & 11 \\
11 & & 11 & 8 & 11 & 3 & 11 & 11 & 11 & 2 & 11 & 5 & 11 & 11 & 7 & 0 & 11 & 7 & \\
9 & 1 & & & 2 & 11 & 3 & 5 & & 11 & 5 & 11 & 6 & & & 11 & 0 & 11 & 3 \\
11 & 11 & 11 & 2 & 11 & 6 & 11 & 11 & & 6 & 11 & 1 & 11 & 11 & & 7 & 11 & 0 & \\
6 & 3 & 6 & & & 11 & 2 & 4 & 3 & 11 & 4 & 11 & & 6 & 11 & & 3 & & 0
\end{bmatrix}$$

we get the sparse Robinsonian realizer, i.e. a permutation among all the permutations fulfilling the Robinson property for all sets of completely specified triples with respect to the permutation)

$$\begin{bmatrix}
0 & 1 & 1 & 2 & 3 & & 6 & 6 & 6 & & 9 & 9 & 11 & 11 & 11 & & 11 & 11 & \\
1 & 0 & & 2 & 3 & 3 & 5 & 5 & 6 & & & 9 & & & 11 & 11 & 11 & 11 & 11 \\
1 & & 0 & 2 & 3 & 3 & 5 & 5 & 6 & 9 & 9 & 9 & 11 & 11 & & 11 & 11 & 11 & 11 \\
2 & 2 & 2 & 0 & & & 4 & 4 & & 6 & 6 & 6 & 11 & 11 & 11 & & 11 & 11 & 11 \\
3 & 3 & 3 & & 0 & 2 & 4 & 4 & & 6 & 6 & 6 & & 11 & & 11 & 11 & 11 & \\
 & 3 & 3 & & 2 & 0 & 4 & 4 & 4 & & 6 & 6 & 11 & 11 & 11 & 11 & & 11 & 11 \\
6 & 5 & 5 & 4 & 4 & 4 & 0 & 1 & & 4 & 6 & & 11 & & 11 & 11 & 11 & 11 & 11 \\
6 & 5 & 5 & 4 & 4 & 4 & 1 & 0 & 2 & 4 & 6 & 6 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
6 & 6 & 6 & & & 4 & & 2 & 0 & 4 & 6 & 6 & & 11 & 11 & 11 & 11 & & 11 \\
 & & 9 & 6 & 6 & & 4 & 4 & 4 & 0 & & & 11 & 11 & 11 & 11 & & 11 & 11 \\
9 & & 9 & 6 & 6 & 6 & 6 & 6 & 6 & & 0 & 2 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
9 & 9 & 9 & 6 & 6 & 6 & & 6 & 6 & & 2 & 0 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
11 & & 11 & 11 & & 11 & 11 & 11 & & 11 & 11 & 11 & 0 & 1 & 2 & 8 & 8 & & 8 \\
11 & & 11 & 11 & 11 & 11 & & 11 & 11 & 11 & 11 & 11 & 1 & 0 & & 2 & 6 & 7 & 7 \\
11 & 11 & & 11 & & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 2 & & 0 & 1 & 6 & 6 & 7 \\
 & 11 & 11 & & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 8 & 2 & 1 & 0 & 5 & 5 & 5 \\
11 & 11 & 11 & 11 & 11 & & 11 & 11 & 11 & & 11 & 11 & 8 & 6 & 6 & 5 & 0 & 1 & 3 \\
11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & & 11 & 11 & 11 & & 7 & 6 & 5 & 1 & 0 & 2 \\
 & 11 & 11 & 11 & & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 8 & 7 & 7 & 5 & 3 & 2 & 0
\end{bmatrix}$$

Renaming upper left corner from 1 to 12 we observe a chordless four cycle $\{1,2,6,3\}$, an obstruction to interval graph; yet, as a special case of Monge property, in antimatroid sense it should.

After first alternating front/rear insertion sweep (under $\min$ / $\max$ dynamic thresholding), starting with the identity, we get the P-nodes

$$\begin{cases} \begin{bmatrix} 19 & 7 & 17 & 9 & 2 & 5 & 11 & 8 & 14 & 13 & 3 & 1 \end{bmatrix} \mapsto 9 \\ \begin{bmatrix} 4 & 18 & 16 & 6 & 15 & 12 & 10 \end{bmatrix} \mapsto 7 \end{cases}$$

$$\begin{bmatrix}
0 & 2 & 3 & 3 & 3 & & 4 & 4 & 6 & & 6 & 6 & & & & 11 & 11 & 11 & 11 \\
2 & 0 & 3 & 3 & & & 4 & 4 & & 4 & 6 & 6 & 11 & 11 & 11 & & 11 & 11 & 11 \\
3 & 3 & 0 & & 1 & 2 & 5 & 5 & & 6 & & 9 & & 11 & 11 & 11 & & 11 & 11 \\
3 & 3 & & 0 & 1 & 2 & 5 & 5 & 9 & 6 & 9 & 9 & 11 & & 11 & 11 & 11 & 11 & 11 \\
3 & & 1 & 1 & 0 & 2 & 6 & 6 & & 6 & 9 & 9 & 11 & 11 & & 11 & 11 & & 11 \\
 & & 2 & 2 & 2 & 0 & 4 & 4 & 6 & & 6 & 6 & 11 & 11 & 11 & 11 & 11 & & 11 \\
4 & 4 & 5 & 5 & 6 & 4 & 0 & 1 & 4 & 2 & 6 & 6 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
4 & 4 & 5 & 5 & 6 & 4 & 1 & 0 & 4 & & 6 & & 11 & 11 & 11 & 11 & & 11 & 11 \\
6 & & & 9 & & 6 & 4 & 4 & 0 & 4 & & & 11 & 11 & 11 & & 11 & 11 & 11 \\
 & 4 & 6 & 6 & 6 & & 2 & & 4 & 0 & 6 & 6 & & 11 & 11 & 11 & 11 & 11 & \\
6 & 6 & & 9 & 9 & 6 & 6 & 6 & & 6 & 0 & 2 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
6 & 6 & 9 & 9 & 9 & 6 & 6 & & & 6 & 2 & 0 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
 & 11 & & 11 & 11 & 11 & 11 & 11 & 11 & & 11 & 11 & 0 & 2 & 8 & 8 & 1 & 8 & \\
 & 11 & 11 & & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 2 & 0 & 7 & 6 & & 1 & 6 \\
 & 11 & 11 & 11 & & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 8 & 7 & 0 & 3 & 7 & 5 & 2 \\
11 & & 11 & 11 & 11 & 11 & 11 & 11 & & 11 & 11 & 11 & 8 & 6 & 3 & 0 & 6 & 5 & 1 \\
11 & 11 & & 11 & 11 & 11 & 11 & & 11 & 11 & 11 & 11 & 1 & & 7 & 6 & 0 & 2 & 7 \\
11 & 11 & 11 & 11 & & & 11 & 11 & 11 & 11 & 11 & 11 & 8 & 1 & 5 & 5 & 2 & 0 & 5 \\
11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & & 11 & 11 & & 6 & 2 & 1 & 7 & 5 & 0
\end{bmatrix}$$

Again, notice that Chepoi and Fichet (1997) no longer applies on threshold graphs with missing entries, for example on lower right block after renaming from 1 to 7, it yields:

$$S = S_8(2) = S_8(3) = S_8(4) = \{6\}, \quad C = \{2,3,4\}, \quad L = \{1,5,7\}$$

$$\begin{cases} S^- = \emptyset, \quad L^- = \{1\} \\ S^+ = \{6\}, \quad L^+ = \{5,7\} \\ S^= = \emptyset \end{cases}$$

but the ordering $\{1\}, \{2,3,4\}, \{5,7\}, \{6\}$ fails since L^- and C have to be mixed due to missing entry d_{17} for a compatible ordering $3,1,2,4,5,7,6$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 7 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 & 5 & 6 & 7 & \\ 1 & 0 & 3 & 5 & 6 & 6 & 8 \\ 2 & 3 & 0 & 5 & 7 & 7 & 8 \\ 5 & 5 & 5 & 0 & 1 & 2 & 8 \\ 6 & 6 & 7 & 1 & 0 & & 2 \\ 7 & 6 & 7 & 2 & & 0 & 1 \\ & 8 & 8 & 8 & 2 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \\ 5 \\ 7 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 3 & 5 & 7 & 7 & 8 \\ 2 & 0 & 1 & 5 & 6 & 7 & \\ 3 & 1 & 0 & 5 & 6 & 6 & 8 \\ 5 & 5 & 5 & 0 & 1 & 2 & 8 \\ 7 & 6 & 6 & 1 & 0 & & 2 \\ 7 & 7 & 6 & 2 & & 0 & 1 \\ 8 & & 8 & 8 & 2 & 1 & 0 \end{bmatrix}$$

5.2.2 Sorting Refinement Rule

Consider the upper left P-node in above example, and sort the rows by decreasing maximum value, say $[3,4,5,9,11,12,7,8,1,2,6,10]$. Then, the sorting rule adapted to missing entries refines as (trivial pivoting skipped)

[3, [5], [6], [1, 2], [7, 8], [10], [12], [4, 9, 11]]
[3, 5, [4], [6], [1, 2], [7, 8], [10], [12, 11], [9]]
[3, 5, 4, [6], [1, 2], [7, 8], [10], [12, 11, 9]]
[3, 5, 4, 6, [1, 2], [7, 8], [10], [12, 11, 9]]
[3, 5, 4, 6, 1, [2], [7, 8], [10], [12, 11, 9]]
[3, 5, 4, 6, 1, 2, [7, 8], [10], [12, 11, 9]]
[3, 5, 4, 6, 1, 2, 7, [8], [10], [9], [12, 11]]

$$\begin{bmatrix}
0 & 1 & & 2 & 3 & 3 & 5 & 5 & 6 & & 9 & 0 \\
1 & 0 & 1 & 2 & 3 & & 6 & 6 & 6 & & 9 & 9 \\
 & 1 & 0 & 2 & 3 & 3 & 5 & 5 & 6 & 9 & 9 & 9 \\
2 & 2 & 2 & 0 & & & 4 & 4 & & 6 & 6 & 6 \\
3 & 3 & 3 & & 0 & 2 & 4 & 4 & & 6 & 6 & 6 \\
3 & & 3 & & 2 & 0 & 4 & 4 & 4 & & 6 & 6 \\
5 & 6 & 5 & 4 & 4 & 4 & 0 & 1 & 2 & 4 & 6 & 6 \\
5 & 6 & 5 & 4 & 4 & 4 & 1 & 0 & & 4 & & 6 \\
6 & 6 & 6 & & & 4 & 2 & & 0 & 4 & 6 & 6 \\
 & & 9 & 6 & 6 & & 4 & 4 & 4 & 0 & & \\
9 & 9 & 9 & 6 & 6 & 6 & 6 & & 6 & & 0 & 2 \\
 & 9 & 9 & 6 & 6 & 6 & 6 & 6 & 6 & & 2 & 0
\end{bmatrix}$$

and second sweep as

[11, [12], [10, 8, 7, 2, 1, 6], [4, 5], [9, 3]]
[11, 12, [10, 8, 7, 2, 1, 6], [4, 5, 3], [9]]
[11, 12, 10, [7], [8, 2, 1, 6, 9], [4, 5, 3]]
[11, 12, 10, 7, [8], [2, 1, 6, 9], [4, 3], [5]]
[11, 12, 10, 7, 8, [2, 1, 6, 9], [4, 3], [5]]
[11, 12, 10, 7, 8, 2, [1, 6, 9], [4, 3], [5]]
[11, 12, 10, 7, 8, 2, [1], [6, 9], [4, 3], [5]]

$$\begin{bmatrix}
0 & 2 & 6 & 6 & 6 & 6 & 6 & 6 & & 9 & & 9 \\
2 & 0 & 6 & 6 & & 6 & 6 & 6 & & 9 & 9 & 9 \\
6 & 6 & 0 & 2 & & 4 & & & 4 & 6 & 6 & 6 \\
6 & 6 & 2 & 0 & 1 & 4 & 4 & 4 & 4 & 5 & 5 & 6 \\
6 & & & 1 & 0 & 4 & 4 & 4 & 4 & 5 & 5 & 6 \\
6 & 6 & 4 & 4 & 4 & 0 & 2 & & & 3 & 3 & 0 \\
6 & 6 & & 4 & 4 & 2 & 0 & & 6 & 3 & 3 & 3 \\
6 & 6 & & 4 & 4 & & & 0 & 6 & 2 & 2 & 2 \\
 & & 4 & 4 & 4 & & 6 & 6 & 0 & 9 & & \\
9 & 9 & 6 & 5 & 5 & 3 & 3 & 2 & 9 & 0 & & 1 \\
 & 9 & 6 & 5 & 5 & 3 & 3 & 2 & & & 0 & 1 \\
9 & 9 & 6 & 6 & 6 & & 3 & 2 & & 1 & 1 & 0
\end{bmatrix}$$

however a third sweep leads to

[5, [3, 4], [6], [1], [8, 7, 10], [12, 11], [9, 2]]
[5, 3, [4], [6], [1, 2], [8, 7], [10], [12, 11], [9]]
[5, 3, 4, [6], [1, 2], [8, 7], [10], [12, 11, 9]]
[5, 3, 4, 6, [1, 2], [8, 7], [10], [12, 11, 9]]
[5, 3, 4, 6, 1, [2], [8, 7], [10], [12, 11, 9]]
[5, 3, 4, 6, 1, 2, [8, 7], [10], [12, 11, 9]]
[5, 3, 4, 6, 1, 2, 8, [7], [10], [9], [12, 11]]

$$\begin{bmatrix}
0 & 1 & 1 & 2 & 3 & & 6 & 6 & 6 & & 9 & 9 \\
1 & 0 & & 2 & 3 & 3 & 5 & 5 & 6 & & 9 & 0 \\
1 & & 0 & 2 & 3 & 3 & 5 & 5 & 6 & 9 & 9 & 9 \\
2 & 2 & 2 & 0 & & & 4 & 4 & & 6 & 6 & 6 \\
3 & 3 & 3 & & 0 & 2 & 4 & 4 & & 6 & 6 & 6 \\
 & 3 & 3 & & 2 & 0 & 4 & 4 & 4 & & 6 & 6 \\
6 & 5 & 5 & 4 & 4 & 4 & 0 & 1 & & 4 & & 6 \\
6 & 5 & 5 & 4 & 4 & 4 & 1 & 0 & 2 & 4 & 6 & 6 \\
6 & 6 & 6 & & & 4 & & 2 & 0 & 4 & 6 & 6 \\
 & & 9 & 6 & 6 & & 4 & 4 & 4 & 0 & & \\
9 & 9 & 9 & 6 & 6 & 6 & & 6 & 6 & & 0 & 2 \\
9 & & 9 & 6 & 6 & 6 & 6 & 6 & 6 & & 2 & 0
\end{bmatrix}$$

a sparse Robinsonian dissimilarity (as expected for recognizing **UI** property in 3 sweeps, Corneil, 2014), Let row r be *congruent* to row s if and only if $d_{rj} = d_{sj}$ for all $j \neq r, s$ (trivially extended to either missing entry); then rows $[1, 2]$, $[3, 4]$ $[7, 8]$ and $[11, 12]$ are *congruent* and we retrieve all 16 sparse Robinsonian matrices with $[5, [3, 4], 6, [1, 2], [8, 7], 10, 9, [12, 11]]$ along their reversals. The lower right P-node in above example, leads to the (single up to reversal) sparse Robinsonian dissimilarity at end of the first sweep with the same sorting rule.

5.3 Dense Robinsonian Failure

For unit interval graphs, as emphasized by Laurent and Seminaroti (2015a), the recognition algorithm simplifies to three standard LexBFS; yet, the extra preprocessing sweep for early detect P-nodes clearly speeds up their whole recognition. In any case, the decision problem relies on the consecutiveness in semi-umbrella structures, either forward for **I** (four sweeps) and both for **UI** (three sweeps). To go beyond the negative answer to the decision problem, we propose to extend the Robinson property as

Definition 5.1 A dissimilarity is $p-$Robinsonian if there exist orderings $\pi_1, \dots \pi_p$ such that for all $1 \leq i < j < k \leq n, \exists \pi_q, \max(d_{\pi_q(i)\pi_q(j)}, d_{\pi_q(j)\pi_q(k)}) \leq d_{\pi_q(i)\pi_q(k)}$

The list $\pi_1, \dots \pi_p$ is called a realizer of $p-$Robinson property. This extension circumvents the quantitative difficulty of approximating a dissimilarity failing the Robinson property (Chepoi et al., 2009), into a more tractable qualitative property around a set of permutations. In this regard, failure of interval property allows to keep the corresponding permutation and to restart the whole interval recognition after removing the semi-umbrellas that are consecutive according to Theorem 3.1. This heuristic terminates in a finite number of restarts so that for all $i < j < k$ the Robinsonian property is fulfilled by at least one permutation. However, it raises the question of finding the minimum number of such restarts for a minimal $p-$Robinsonian dissimilarity.

5.4 Anti-Robinsonian Recognition

Observe that anti-Robinsonian property is invariant by reversal, for all $1 \leq i < j < k \leq n$

$$\max(a_{n+1-k,n+1-j}, a_{n+1-j,n+1-i}) \geq a_{n+1-k,n+1-i} \cdot$$

Therefore, P-node dividing scheme applies; however, it remains local in absence of a cone structure. It exemplifies the observation that recognition suffers from the lack of convexity (piecewise convexity only). It translates into graph setting by the definition of a cocomparability graph as the complement of a comparability graph (namely, admitting a transitive orientation). A realizer of a cocomparability graph extends the notion of a realizer for a permutation graph ($i?j$ iff π_i, π_j are reversed by the permutation π) over a set of permutations instead.

Definition 5.2 A cocomparability ordering of a graph G is an umbrella-free total ordering of the vertices of G.

R. McConnell and J.P. Spinrad (McConnell and Spinrad, 1994) have proved linear $O(m)$ algorithms for producing a cocomparability ordering and D. Corneil improves for every cocomparability graph, there exist a cocomparability ordering which is a LexBFS ordering (see Corneil et al., 1999, for dominating pairs in AT-free graphs) and E. Köhler and L. Mouatadid (Köhler and Mouatadid, 2014) recently found a LexDFS cocomparability ordering. To our knowledge, there is no certifying (finding a realizer as a set of permutations) algorithm for cocomparability recognition that assert a better complexity than $O(nm)$ with n LexBFS sweeps while there is no strong evidence that LexDFS has to be discarded. On the other hand, the growth of the order dimension of the complete graph (Hoşten and Morris, 1999) is slower than $\log(n)$; therefore, $O(nm)$ bound with LexBFS sweeps only, sounds quite large. It leaves open the challenging task of recognition of intimately coupled cocomparability and anti-Robinson properties.

5.5 Lex Ordering Polytope

Apart from the anti-Robinsonian reconition challenge, another instance where both LexDFS and LexBFS are proven useful is planar embedding. Given three points as suspension of the outer face of a planar graph, Schnyder woods consist in $\binom{3}{2}$ orderings whose endpoints have 2 suspension points, i.e. a structure slightly more complicated than invariant by reversal. It may deserve understanding the extensive literature about planar embedding in this respect, since planar convexity looks comparable with (resp. simpler than) piecewise convexity of anti-Robinson property with a realizer of size 2 (resp. > 2). The following comparison between the mixed LexBFS and LexDFS ordering polytope and the metric polytope already shows the higher complexity structure for small size $n = 7$. Recall the ternary neighborhood operator in a graph $G = (V, E)$ is $i?j : k$ iff $(i, j) \in E$ together with $(i, k) \notin E$.

Proposition 5.1 [BFS/DFS elimination ordering (Corneil et al., 2016)] *For an ordering σ and all $j <_\sigma k <_\sigma l \in V$*

Table 1. #vertices in the small sized *Lex ordering/Metric* polytopes

	Lex Ordering		Metric	
	#vertices		#vertices	
size	integral	fractional	integral	fractional
4	1	0	6	0
5	9	3	26	10
6	127	554	116	746
7	1456	1056863	642	173788

$$\textit{if } j?l : k \quad \textit{then } \exists i <_\sigma j \quad \textit{s.t.} \quad i?k \tag{BFS}$$

$$\textit{if } i?l : k \quad \textit{then } \exists i <_\sigma j <_\sigma k \quad \textit{s.t.} \quad j?k \tag{DFS}$$

Proposition 5.2 [LexBFS/DFS elimination ordering (Corneil et al., 2016)] *For an ordering σ and all $j <_\sigma k <_\sigma l \in V$ l*

$$\textit{if } j?l : k \quad \textit{then } \exists i <_\sigma j \quad \textit{s.t.} \quad i?k : l \tag{LexBFS}$$

$$\textit{if } i?l : k \quad \textit{then } \exists i <_\sigma j <_\sigma k \quad \textit{s.t.} \quad j?k : l \tag{LexDFS}$$

In analogy with the metric polytope, consider the set of non-linear inequalities associated with the BFS/DFS elimination ordering

$$x_{jl}(1 - x_{jk}) \leq x_{ik}, \quad \text{for all } i < j < k < l$$
$$x_{il}(1 - x_{ik}) \leq x_{jk}, \quad \text{for all } i < j < k < l\,.$$

Linearizing the left hand side is revealed as less accurate than the LexBFS/ LexDFS elimination ordering; however linearized per se, we have

$$x_{ik} + x_{il} + x_{jk} + x_{jl} \leq 2, \quad \text{for all } i < j < k < l\,.$$

Despite the prohibitive n^4 size, it deserves studying the polytope obtained along with the triangle inequalities

$$x_{ik} - x_{ij} - x_{jk} \leq 0, \quad \text{for all } i < j < k$$

in the same way as Laurent (1996). However, the vertex enumeration (Avis and Fukuda, 1992) blows up very quickly (see Table 1). For size 7, they exhibit (by one order of magnitude) more fractional vertices than the corresponding metric polytope, showing the strength of lexicographic searches, either LexBFS or LexDFS.

5.6 Concluding Remarks

In this article, we review known algorithms for the interval graph recognition since it is at the heart of the decision problem for dense Robinsonian matrices recognition. The four LexBFS sweeps algorithm allows to go beyond a failure in regard to interval graph property, by computing a local realizer for a set of p intervals at each point; it leads to a natural extension to $p-$Robinsonian matrices as a set of p permutations such that for all $1 \leq i < j < k \leq n$, at least one permutation is compatible with the anti-ultrametric inequality. For sparse matrices, the underlying connection between Robinsonian matrices and interval graph property of threshold graphs collapses; yet, it holds under antimatroid framework and it is conjectured that LexBFS antimatroid tie breaking rule exists to yield a four sweeps sparse recognition algorithm. The dense antiRobinson property is proven harder to address due to its intimate connection with cocomparability graphs on the one hand, and its piecewise convex property, on the other hand. Compared with the corresponding metric polytope, the lexicographic polytope (mixing BFS and DFS) has much more fractional vertices and argues in favor of lexicographic searches over optimization formulations for solving the recognition problems. Challenging tasks to unify sparse Robinsonian, $p-$Robinsonian and dense antiRobinsonian matrices remain open: the sparse Robinsonian task looks reachable with four LexBFS sweeps while the others may require mixed lexicographic searches to certify a realizer.

References

AVIS, D., and FUKUDA, K. (1992), “A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra", *Discrete and Computational Geometry, 8(3)*, 295–313.

BERRY, A., and BORDAT, J.-P. (1998), “Separability Generalizes Dirac’s Theorem", *Discrete and Applied Mathematics, 84(1-3)*, 43–53.

BOOTH, K.S., and LUEKER, G.S. (1976), “Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms", *Journal of Computer and System Sciences, 13(3)*, 335–379.

BURKARD, R.E., ÇELA, E., ROTE, G., and WOEGINGER, G.J. (1998), “The Quadratic Assignment Problem with a Monotone Anti-Monge and a Symmetric Toeplitz Matrix: Easy and Hard Cases", *Mathematical Programming, 82(1-2, Ser. B)*, 125–158.

ÇELA, E., DEINEKO, V.G., and WOEGINGER, G.J. (2012), “Another Well-Solvable Case of the QAP: Maximizing the Job Completion Time Variance", *Operations Research Letters, 40(5)*, 356–359.

CHEPOI, V., and FICHET, B. (1997), “Recognition of Robinsonian Dissimilarities", *Journal of Classification, 14(2)*, 311–325.

CHEPOI, V., FICHET, B., and SESTON, M. (2009), “Seriation in the Presence of Errors: NP-Hardness of l_∞-Fitting Robinson Structures to Dissimilarity Matrices", *Journal of Classification, 26(3)*, 279–296.

CORNEIL, D.G. (2004), “A Simple 3-Sweep LBFS Algorithm for the Recognition of Unit Interval Graphs", *Discrete Applied Mathematics, 138(3)*, 371–379.

CORNEIL, D.G., OLARIU, S., and STEWART, L. (1999), “Linear Time Algorithms for Dominating Pairs in Asteroidal Triple-Free Graphs", *SIAM Journal on Computing, 28(4)*, 1284–1297.

CORNEIL, D.G., OLARIU, S. and STEWART, L. (2009/10), “The LBFS Structure and Recognition of Interval Graphs", *SIAM Journal on Discrete Mathematics, 23(4)*, 1905–1953.

CORNEIL, D.G., DUSART, J., HABIB, M., MAMCARZ, A., and DE MONTGOLFIER, F. (2016), “A Tie-Break Model for Graph Search", *Discrete Applied Mathematics, 199*, 89–100.

DAHLHAUS, E. (1993), “Fast Parallel Recognition of Ultrametrics and Tree Metrics", *SIAM Journal on Discrete Mathematics, 6(4)*, 523–532.

DIETRICH, B.L. (1990), “Monge Sequences, Antimatroids, and the Transportation Problem with Forbidden Arcs", *Linear Algebra and Its Applications, 139*, 133–145.

FORTIN, D., and RUDOLF, R. (1998), “Weak Monge Arrays in Higher Dimensions", *Discrete Mathematics, 189(1-3)*, 105–115.

HABIB, M., MCCONNELL, R., PAUL, C., and VIENNOT, L. (2000), “Lex-BFS and Partition Refinement, with Applications to Transitive Orientation, Interval Graph Recognition and Consecutive Ones Testing", *Theoretical Computer Science, 234(1-2)*, 59–84.

HOŞTEN, S., and MORRIS, W.D. Jr. (1999), “The Order Dimension of the Complete Graph", *Discrete Mathematics, 201(1-3)*, 133–139.

HUBERT, L., and ARABIE, P. (1994), “The Analysis of Proximity Matrices Through Sums of Matrices Having (Anti)-Robinson Forms", *British Journal of Mathematical and Statistical Psychology, 47*, 1–40.

KLINZ, B., RUDOLF, R., and WOEGINGER, G.J. (1995), “On the Recognition of Permuted Bottleneck Monge Matrices", *Discrete Applied Mathematics, 63(1)*, 43–74.

KÖHLER, E., and MOUATADID, L. (2014), “Linear Time LexDFS on Cocomparability Graphs", in *Algorithm Theory—SWAT 2014*, Vol. 8503, *Lecture Notes in Computer Science*, Springer, pp. 319–330.

LAURENT, M. (1996), “Graphic Vertices of the Metric Polytope", *Discrete Mathematics, 151(1-3)*, 131–153.

LAURENT, M., and SEMINAROTI, M. (2015a), *A Lex-BFS-Based Recognition Algorithm for Robinsonian Matrices*, Springer International Publishing, pp. 325–338.

LAURENT, M., and SEMINAROTI, M. (2015b), “The Quadratic Assignment Problem is Easy for Robinsonian Matrices with Toeplitz Structure", *Operations Research Letters, 43(1)*, 103–109.

LAURENT, M., and SEMINAROTI, M. (2016), “Similarity-First Search: A New Algorithm with Application To Robinsonian Matrix Recognition", *CoRR*, abs/1601.03521, http://arxiv.org/abs/1601.03521.

LEKKERKERKER, C.G., and BOLAND, J.C. (1962/1963), “Representation of a Finite Graph by a Set of Intervals on the Real Line", *Fundamenta Mathematicae, 51*, 45–64.

LI, P., and WU, Y. (2014), “A Four-Sweep LBFS Recognition Algorithm for Interval Graphs", *Discrete Mathematics and Theoretical Computer Science, 16(3)*, 23–50.

MCCONNELL, R.M., and SPINRAD, J.P. (1994), “Linear-Time Modular Decomposition and Efficient Transitive Orientation of Comparability Graphs", in *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994)*, New York: ACM, pp. 536–545.

PRÉA, P., and FORTIN, D. (2014), “An Optimal Algorithm to Recognize Robinsonian Dissimilarities", *Journal of Classification, 31(3)*, 351–385.

ROBINSON, W. (1951), “A Method for Chronologically Ordering Archaeological Deposits", *American Antiquity, 16(4)*, 293–301.

RUDOLF, R., and WOEGINGER, G.J. (1995), “The Cone of Monge Matrices: Extremal Rays and Applications", *ZOR—Mathematical Methods of Operations Research, 42(2)*, 161–168.

TSEVENDORJ, I. (2001), “Piecewise-Convex Maximization Problems: Global Optimality Conditions", *Journal of Global Optimization, 21(1)*, 1–14.

XU, S.-J., LI, X., and LIANG, R. (2013), “Moplex Orderings Generated by the LexDFS Algorithm", *Discrete Applied Mathematics, 161(13-14)*, 2189–2195.