Laboratoire Bordelais de Recherche en Informatique
UMR 5800 - Université Bordeaux I, 351, cours de la Libération,
33405 Talence CEDEX, France

Research Report RR-1290-03

Small Universal Distance Matrices

by Cyril Gavoille and Christophe Paul

February, 2003

Small Universal Distance Matrices

Cyril Gavoille* Christophe Paul†

Abstract

In this paper, we propose to associate to any graph family $\mathcal{F}$ the following complexity measure: the dimension of the smallest *universal distance matrix* for $\mathcal{F}$. Given n, an universal distance matrix for $\mathcal{F}$ is a square matrix $\mathcal{U}_n$ containing the distance matrix of every (unweighted) n-vertex graph of $\mathcal{F}$ as induced sub-matrix, i.e., a matrix obtained by selecting in $\mathcal{U}_n$ a same subset of row and of column indices.

We show that the family of interval graphs has an universal distance matrix of dimension $\mathcal{O}(n^5)$, and we prove a lower bound of $\Omega(n^3/\log^4 n)$ on the dimension of such matrix. Previous upper bound on the dimension of the smallest matrix was $n^{\mathcal{O}(\log n)}$. Our upper bound is obtained by devising a distance labeling scheme with $\mathcal{O}(\log n)$ bit labels for interval graphs and circular-arc graphs. Moreover, the set of all the labels is constructible in $\mathcal{O}(n)$ time if the interval representation of the graph is given and sorted. As a byproduct we give a new an simpler $\mathcal{O}(n)$ space data structure computable after $\mathcal{O}(n)$ preprocessing time, and supporting constant worst-case time distance queries for interval graphs and circular-arc graphs.

keywords: parallel and distributed computing, parallel data structure, distance labeling, implicit representation, distance matrices, interval graphs, circular-arc graphs

1 Introduction

There are several applications, like phylogeny, whose the fundamental questions are related to distances in graphs. What does look distances in a graph taken from a given graph family $\mathcal{F}$? Is a distance matrix M the metric of graph taken from $\mathcal{F}$? For instance, trees have the famous 4-point condition that allows to decide quickly whether a distance matrix M is the metric of a tree or not [ABF+96]. However, it seems difficult to defined a general framework to express such conditions for any graph family. An also interesting problem is to compare several graph metrics by comparing some complexity measures defined on the graph families themselves. It seems natural to say, for instance, that the metric of cubic

*Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I. *gavoille@labri.fr*

†Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier. *paul@lirmm.fr*

graphs has a higher complexity than the metric of trees. Intuitively, distances in a cubic n-vertex graph might look "much random" than distances in an n-vertex tree.

In this paper, we propose to associate to any family $\mathcal{F}$ the following complexity measure: the dimension of the smallest universal distance matrix for $\mathcal{F}$. Given n, an *universal distance matrix* for $\mathcal{F}$ is a square matrix $\mathcal{U}_n$ containing the distance matrix of every (unweighted) n-vertex graph of $\mathcal{F}$ as induced sub-matrix. A matrix $B = (b_{i,j})_{1 \leqslant i,j \leqslant p}$ is an induced sub-matrix of a matrix $A = (a_{i,j})_{1 \leqslant i,j \leqslant q}$ if there exists a sequence $(s_1, \ldots, s_p)$ of indices such that $b_{i,j} = a_{s_i,s_j}$, for all $i, j \in \{1, \ldots, p\}$. Intuitively, a small universal distance matrix for $\mathcal{F}$ indicates that many distance patterns repeat in many different graphs of $\mathcal{F}$. As we will see trees have a universal distance matrix of dimension $2^{\Theta(\log^2 n)}$ whereas any universal distance matrix for cubic graphs must be of dimension at least $2^{\Omega(\sqrt{n})}$.

An important observation is that a universal distance matrix $\mathcal{U}_n$ for $\mathcal{F}$ can be used to label the vertices of any n-vertex graph G of $\mathcal{F}$ by an integer such a way the distance between any two vertices of G can be computed from the their label, without any global information on G. Indeed, as the distance matrix of G appears as an induced sub-matrix of $\mathcal{U}_n$, the row index of $\mathcal{U}_n$ can be used to label the vertex associated to the row of the distance matrix of G. So the dimension of $\mathcal{U}_n$ gives an upper bound on the number of different labels needed to label every vertex of every n-vertex graph of $\mathcal{F}$.

Labeling of a graph with such implicit distance computation capabilities is called *distance labeling scheme*. Formally [Pel00b], a distance labeling scheme for a graph family $\mathcal{F}$ is a pair $\langle L, f \rangle$ of functions such that $L(v, G)$ is a binary label associated to the vertex v in the graph G, and such that $f(L(x, G), L(y, G))$ returns the distance between the vertices x and y in the graph G, for all x, y of G and every $G \in \mathcal{F}$. The labeling scheme is said an $\ell(n)$-distance labeling if for every n-vertex graph $G \in \mathcal{F}$, the length of the labels are no more than $\ell(n)$ bits.

A labeling scheme using short labels is clearly a desirable property for a graph, especially in the framework of parallel and distributing computing where individual processor element of a network want to communicate with its neighbors but has not enough local memory resources to store all the underlying topology of the network. Schemes providing compact labels play an important role for localized parallel and distributed data structures (see [GP01b] for a survey).

1.1 Related works

In this framework [Pel00a] introduced *informative labeling schemes* for graphs and thereby captured a whole set of already known results such as implicit representation or adjacency labeling [AR02b, KNR92, Spi02] whose objective is only to decide adjacency between two vertex labels, compact routing [Gav01, TZ01b] whose objective is to provide the first edge on a near-shortest path between the source and the destination, nearest common ancestor and other related functions for trees [ABR03, AGKR02, AKM01, KMS02], flow and connectivity [KKP02].

A first set of results about distance labeling can be founded in [GPPR01]. It is shown, for instance, that general n-vertex graphs enjoy an optimal $\Theta(n)$-distance labeling, and trees an optimal $\Theta(\log^2 n)$-distance labeling. It is also proved a lower bound of $\Omega(\sqrt{n}\,)$ on the label length for bounded degree graphs, and of $\Omega(n^{1/3})$ for planar graphs. There are still intriguingly gap between upper and lower bounds for these two latter results. The upper bound for planar graphs is $\mathcal{O}(\sqrt{n}\log n)$, coming from a more general result about graphs having small separators. Related works concern distance labeling schemes in dynamic tree networks [KPR02], and approximate distance labeling schemes [GKK+01, Tho01, TZ01a].

Several efficient schemes have been designed for specific graph families: interval and permutation graphs [KKP00], distance hereditary graphs [GP01a], bounded tree-width and more generally bounded clique-width graphs [CV01], all support an $\mathcal{O}(\log^2 n)$-distance labeling scheme. Excepted for the two first families, schemes are known to be optimal because theses families include trees.

1.2 Our results

An interval graph is a graph whose vertices are intervals of the real line and whose edges are defined by the intersecting intervals. Trees are not in general interval graphs. Thus the $\Omega(\log^2 n)$ lower bound of [GPPR01] does not apply. On the other hand, no hereditary graph family (i.e., a family supporting vertex deletion) is known to support an $o(\log^2 n)$-distance labeling scheme[1]. Like trees and several other families (bounded tree-width, bounded clique-width, bounded genus graphs, etc.) interval graphs support $\mathcal{O}(\log n)$ bit adjacency labeling scheme. Moreover, there are much more interval graphs than all the above mentioned graphs. Indeed, trees, bounded tree-width, bounded clique-width, and bounded genus graphs possesses only $2^{\mathcal{O}(n)}$ unlabeled n-vertex graphs. As we will see later there are $2^{\Omega(n \log n)}$ unlabeled interval graphs. All these remarks seem to plead in Katz-Katz-Peleg's favor with their $\mathcal{O}(\log^2 n)$-distance labeling scheme [KKP00].

Surprisingly, we show that the interval graph family has an universal distance matrix of dimension $O(n^5)$, and thus enjoys an $\mathcal{O}(\log n)$-distance labeling scheme. This improves the best previous scheme by a multiplicative factor of $\mathcal{O}(\log n)$. We also present lower bounds, proving that every universal distance matrix for interval graphs is of dimension at least $\Omega(n^3/\log^4 n)$.

Moreover, once the labels have been assigned, the distance computation from the labels takes a constant number of additions and comparisons on $O(\log n)$ bit integers. Even more interesting, the preprocessing time to set all the labels runs optimally in $\mathcal{O}(n)$ time, once the sorted list of intervals of the graph is done. Our scheme extends to circular-arc graphs, a natural generalization of interval graphs, where the same bounds apply.

At this step, it is worth to remark that any $\ell(n)$-distance labeling scheme on a family

[1] It is not difficult to construct a family of diameter two graphs whose adjacency can be decided with $\mathcal{O}(\log n)$ bit labels (some bipartite graphs for instance), so supporting an $\mathcal{O}(\log n)$ distance labeling scheme. However, "diameter two" is not a hereditary property.

$\mathcal{F}$ converts trivially into a non-distributed data structure for $\mathcal{F}$ of $\mathcal{O}(\ell(n)\, n/\log n)$ space supporting distance queries within the same time complexity, being assumed that a cell of space can store $\Omega(\log n)$ bits of data. Therefore, our result implies that interval graphs have a $\mathcal{O}(n)$ space data structure, constructible in $\mathcal{O}(n)$ time, supporting constant time distance queries.

This latter formulation implies the result of Chen et al. [CLSS98]. However, we highlight that both approaches differ in essence. Their technique consists in building a one-to-one mapping from the vertices of the input graph to the nodes of a rooted tree, say T. Then, distances are computed as follows. Let $l(v)$ be the level of v in T (i.e., the distance from the root), and let $a(i, v)$ be the i-th ancestor of v (i.e., the i-th node on the path from v to the root). The distance d between x and y is computed by: if $l(x) > l(y) + 1$ then $d = l(x) - l(y) - 1 + d^\star(z, x)$ where $z = a(l(x) - l(y) - 1, x)$, and where $d^\star(z, x)$ is the distance between two nodes whose levels differ by at most 1. The distance $d^\star$ is $1, 2$ or 3 depending on the interval representation of the involved vertices. This scheme is mainly based on the efficient parallel implementation of *level ancestor queries* on trees (to compute z) given by Berkman and Vishkin [BV94]. However, this clever scheme cannot be converted into a distributed data structure for the following reason. As the tree has to support level ancestor queries, it implies that any node, if represented with a local label, can extract any of its ancestors with its level. In particular, x and y can extract from their label their nearest common ancestor and its level, so x and y can compute their distance in T. By the lower bound of [GPPR01], this cannot be done in less than $\Omega(\log^2 n)$ bit labels. So, access to a global data structure is inherent to the Chen et al. [CLSS98] approach.

1.3 Outline of the paper

Let us sketch our technique. The key of our scheme is a reduction to proper interval graphs, namely the family of graphs having an interval representation with only proper intervals, i.e., with no strictly included intervals. We carefully add edges to the input graph (by extended the length of non-proper intervals) up to obtain a proper interval graph. Distances in the original graph can be retrieved from the label of two vertices of the proper interval graph and from the original interval (see Section 4).

Therefore, the heart of our scheme is based on an efficient labeling of proper interval graphs, and it is presented in Section 3. First, an integer $\lambda(x)$ is associated to every vertex x with the property that the distance d between x and y is: $d = \lambda(y) - \lambda(x) + \delta(x, y)$, if $\lambda(y) \geqslant \lambda(x)$, and where $\delta(x, y) = 0$ or 1. An important step is that the binary relation δ has the structure of a 2-dimensional poset. And we show how to assigned in $\mathcal{O}(n)$ time a number $\pi(x)$ to every x such that $\delta(x, y) = 1$ iff $\pi(x) > \pi(y)$. It turns out a $2\log n$-distance labeling for proper interval graphs, and a $5\log n$-distance labeling for general interval graphs[2].

In Section 5 we show how to reduce distance labeling scheme on circular-arc graphs

[2]In this paper all the logarithms are in based two.

to interval graphs by "unrolling" twice the input circular-arc graph. Distances can be recovered by doubling the label length.

In Section 2, we also prove lower bounds on the label length and on the dimension of universal distance matrices. A byproduct of our counting argument, used for the lower bounds, is a new asymptotic on the number of n-vertex interval graphs, solving an open problem of Hanlon [Han82].

Relationship between universal distance matrices and distance labeling schemes is presented in Section 2.

2 Universal Distance Matrix and Lower Bounds

For any graph family $\mathcal{F}$, let $\mathcal{F}_n$ denote the set of graphs of $\mathcal{F}$ having at most n vertices. For all vertices x, y of a graph G, we denote by $\text{dist}_G(x, y)$ distance between x and y in G.

As said in the introduction, distance labeling scheme and universal distance matrix are two ways of defining the same concept. More precisely:

Proposition 2.1 *If $\mathcal{F}$ enjoys an $\ell(n)$-distance labeling scheme, then $\mathcal{F}$ has a universal distance matrix of dimension $2^{\ell(n)+1}-1$. If $\mathcal{F}$ has a universal distance matrix of dimension $d(n)$, then $\mathcal{F}$ enjoys a $\lceil \log(d(n)+1) - 1 \rceil$-distance labeling scheme.*

Proof. The proof technique is similar to the proof given by [KNR92] for relationship between adjacency schemes and *induced universal graphs* [AR02a, Chu90].

For every non-null integer i, let $s(i)$ denote the i-th binary string in the lexicographic order. We denote by s^{-1} the inverse function of s.

A universal distance matrix M for $\mathcal{F}$ can be defined from any distance labeling scheme $\langle L, f \rangle$ for $\mathcal{F}$ as follows: $M_{i,j} = f(s(i), s(j))$ if there exists a graph $G \in \mathcal{F}_n$ with two vertices u and v such that $L(u, G) = s(i)$ and $L(v, G) = s(j)$; and $M_{i,j} = 0$ otherwise.

For every $G \in \mathcal{F}_n$, one can associate the subset $S_G = \{s^{-1}(L(u, G)) \mid u \in V(G)\}$. By construction, for all $i, j \in S_G$, $M_{i,j} = f(s(i), s(j))$ that is the distance in G between the vertices labeled $s(i)$ and $s(j)$. So M contains as induced sub-matrix the distance matrix of every $G \in \mathcal{F}_n$: M is universal. Moreover, its dimension is no more than the higher integer contained in a subset S_G. This is the number of binary strings that are of length at most $\ell(n)$, i.e., $2^{\ell(n)+1} - 1$.

Now, if M is a universal distance matrix of dimension $d(n)$ for $\mathcal{F}$, then a distance labeling scheme $\langle L, f \rangle$ can be constructed as follows: if the distance matrix of G is the induced sub-matrix of M with sequence of indices S_G, then we label the vertex corresponding to the index $i \in S_G$ by the binary string $s(i)$. The length of this label is the length of $s(i)$. The length is therefore bounded by the length of $s(d(n))$, that is $\lceil \log(d(n)+1) - 1 \rceil$ bits. Now, given two binary labels λ and λ', extracted from the same

graph, one can compute the distance by computing M (this depends on $\mathcal{F}_n$ only), and returning $M_{s^{-1}(\lambda),s^{-1}(\lambda')}$. As the two labels are extracted from the same graph, this value is the distance between the corresponding vertices. □

An application of Proposition 2.1, combined with the bounds of [GPPR01], is that the smallest universal distance matrix for trees is of dimension $2^{\Theta(\log^2 n)}$ and is at least of dimension $2^{\Omega(\sqrt{n})}$ for cubic graphs. Proposition 2.1 will be a tool for proving lower and upper bounds on the dimension of universal distance matrix.

An α-graph, for integer $\alpha \geqslant 1$, is a graph G having a pair of vertices (l, r), possibly with $l = r$, such that l and r are of eccentricity at most α. Let $H = (G_0, G_1, \ldots, G_k)$ be a sequence of α-graphs, and let (l_i, r_i) denote the pair of vertices that defines the α-graph G_i, for $i \in \{0, \ldots, k\}$. For each non-null integer sequence $W = (w_1, \ldots, w_k)$, we denote by H^W the graph obtained by attaching a path of length w_i between the vertices r_{i-1} and l_i, for every $i \in \{1, \ldots, k\}$ (see Fig. 1).

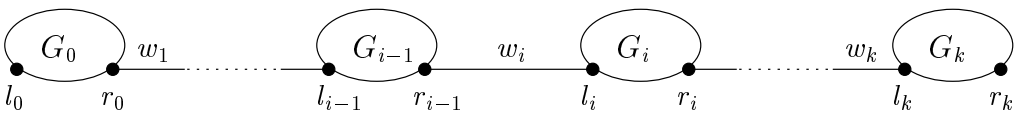

Figure 1: Linking a sequence of α-graphs.

A sub-family $\mathcal{H} \subset \mathcal{F}$ of graphs is *α-linkable* if $\mathcal{H}$ consists only of α-graphs and if $H^W \in \mathcal{F}$ for every graph sequence H of $\mathcal{H}$ and every non-null integer sequence W.

The following lemma shows a lower bound on the length of the labels used by a distance labeling scheme on any graph family having an α-linkable sub-family. The bound is related to the number of labeled graphs contained in the sub-family. As we will see later, the interval graph family supports a large 1-linkable sub-family (we mean large w.r.t. n), and the proper interval graph family supports large 2-linkable sub-family.

Lemma 2.1 *Let $\mathcal{F}$ be any family, and let $F(N)$ be the number of labeled N-vertex graphs of an α-linkable sub-family of $\mathcal{F}$. Then, every distance labeling scheme on $\mathcal{F}_n$ requires a label of length at least $\frac{1}{N}\log F(N) + \log N - 9$, where $N = \lfloor n/(\alpha \log n)\rfloor$.*

Proof. Let us first sketch the proof. We use a sequence H of $\Theta(\log n)$ graphs G_i taken from an arbitrary α-linkable sub-family, each with $\Theta(n/\log n)$ vertices, and spaced with paths of length $\Theta(n/\log n)$. Intuitively, the term $\frac{1}{N}\log F(N)$ measures the minimum label length required to decide whether the distance between any two vertices of a same G_i's is one or two. And the term $\log N$ denotes the minimum label length required to compute the distance between two vertices of consecutive G_i's. The difficulty is to show that some vertices require both information. Indeed, information can be distributed on the vertices in a non trivial way. For instance, the two extremity vertices of a path of length w_i does not require $\log w_i$ bit labels, but only $\frac{1}{2}\log w_i$ bits: each extremity can store one half of the

binary word representing w_i, and merge their label for a distance query. Let us formalize the proof more precisely.

Let p, q, k be non-null integers. Let $\mathcal{W}$ be the set of all the integer sequences $(w_1, \ldots, w_k)$ such that $w_i \in \{1 + (j-1)(2\alpha+1) \mid 1 \leqslant j \leqslant q\}$. We consider $\mathcal{F}'$ be an arbitrary α-linkable sub-family of $\mathcal{F}$. For every $W \in \mathcal{W}$, we denote by $\mathcal{H}_p(W)$ the set of all the graphs H^W where H is a sequence of $k+1$ graphs of $\mathcal{F}'$, each with exactly p vertices (see Fig. 1).

The number of vertices of any graph of $\mathcal{H}_p(W)$ is $(k+1)p + \left(\sum_{i=1}^{k}(w_i - 1)\right) + 1 \leqslant (k+1)p + k(q-1)(2\alpha+1) + 1 \leqslant (k+1)(p + 3\alpha q)$ as $\alpha \geqslant 1$. Since $\mathcal{F}'$ is an α-linkable sub-family of $\mathcal{F}$, each graph H^W belongs to $\mathcal{F}$, and thus, for every $n \geqslant (k+1)(p+3\alpha q)$, $\bigcup_{W \in \mathcal{W}} \mathcal{H}_p(W) \subseteq \mathcal{F}_n$.

Let $\langle L, f \rangle$ be any distance labeling on $\mathcal{F}_n$. We consider the set $\mathcal{L}_W$ of all the $(k+1)$-tuples of labels for all the graphs $H^W \in \mathcal{H}_p(W)$, each tuples being formed by taken a vertex label successively in $G_0, G_1, \ldots, G_k$. More formally,

$$\mathcal{L}_W = \bigcup_{\substack{(G_0,\ldots,G_k)=H \\ H^W \in \mathcal{H}_p(W)}} \{(L(u_0, G_0), \ldots, L(u_k, G_k)) \mid (u_0, \ldots, u_k) \in V(G_0) \times \cdots \times V(G_k)\}$$

Finally, let $\mathcal{L} = \bigcup_{W \in \mathcal{W}} \mathcal{L}_W$.

Claim 2.2 *L assigns on a graph of $\mathcal{F}_n$ a label of length at least $\frac{1}{k+1}\log|\mathcal{L}| - 1$.*

Indeed, assume that all the labels assigned by L on $\mathcal{F}_n$ are of t bits at most. Then, as there are $2^{t+1}-1$ binary strings of length at most t, any set of $(k+1)$-tuples of such labels is of cardinality at most $(2^{t+1}-1)^{k+1}$. $\mathcal{L}$ is a set of $(k+1)$-tuples, thus $|\mathcal{L}| \leqslant (2^{t+1}-1)^{k+1}$, implying that $t \geqslant \log\left(|\mathcal{L}|^{1/(k+1)} + 1\right) - 1 \geqslant \frac{1}{k+1}\log|\mathcal{L}| - 1$.

Claim 2.3 *$|\mathcal{L}_W|^p \geqslant F(p)^{k+1}$.*

At every graph $H^W \in \mathcal{H}_p(W)$, we can associate the label sequence

$$L_H = (\ell_{1,0}, \ldots, \ell_{1,k}, \ell_{2,0}, \ldots, \ell_{2,k}, \ldots, \ell_{p,0}, \ldots, \ell_{p,k}), \quad \text{with } \ell_{i,j} = L(v_i, G_j),$$

composed of all the labels assigned to the all vertices of each G_j of H, the vertices arbitrarily ordered. We have $\bigcup_{H^W \in \mathcal{H}_p(W)} L_H \subseteq (\mathcal{L}_W)^p$. Let us show that $L_H \neq L_{H'}$ for all $H \neq H'$. Indeed, if $H \neq H'$, then two vertices, say v_{i_1} and v_{i_2}, are adjacent in the j-th graph G_j of H and not adjacent in j-th graph G'_j of H (so they are at distance at least two). The distance decoder f must therefore return different values for the corresponding pair of labels, i.e., $f(L(v_{i_1}, G_j), L(v_{i_1}, G_j)) \neq f(L(v_{i_1}, G'_j), L(v_{i_1}, G'_j))$. As both labels, $\ell_{i_1,j}$ and $\ell_{i_2,j}$, are located in the same place in L_H and in $L_{H'}$, $L_H \neq L_{H'}$. It follows that

$$\left| \bigcup_{H^W \in \mathcal{H}_p(W)} L_H \right| \geqslant |\mathcal{H}_p(W)| \ .$$

Since $\mathcal{H}_p(W)$ contains any sequence of $(k+1)$ p-vertex graphs of $\mathcal{F}$, $|\mathcal{H}_p(W)| = F(p)^{k+1}$. Combining the previous inequalities and inclusions, we have $|\mathcal{L}_W|^p \geqslant F(p)^{k+1}$ as claimed.

Claim 2.4 *For all distinct $W, W' \in \mathcal{W}$, $\mathcal{L}_W \cap \mathcal{L}_{W'} = \varnothing$.*

First observe that if W and W' differ at the i-th entry, with $w_i \neq w'_i$, then $[w_i, w_i + 2\alpha] \cap [w'_i, w'_i + 2\alpha] = \varnothing$. Indeed, by construction of the integers of $\mathcal{W}$, $|w_i - w'_i| \geqslant 2\alpha + 1$. We observe that $f(L(u, G_{i-1}), L(v, G_i)) \in [w_i, w_i + 2\alpha]$, for every $i \in \{1, \ldots, k\}$ and for arbitrary $(u, v) \in V(G_{i-1}) \times V(G_i)$. Note also that every tuple of labels $T \in \mathcal{L}_W$ contains a label taken from G_{i-1} and a label from G_i. So, as the ranges of distances are disjoint, if W and W' differ by the i-th entry, then every element $T \in \mathcal{L}_W$ and every element $T' \in \mathcal{L}_{W'}$ must differ by the $(i-1)$-th or the i-th entry, proving the last claim.

From the last two claims, $|\mathcal{L}| \geqslant \sum_{W \in \mathcal{W}} |\mathcal{L}_W|^{1/p} \geqslant F(p)^{(k+1)/p} q^k$. From the first claim, the label length is at least

$$t \;\geqslant\; \frac{1}{p}\log F(p) + \frac{k}{k+1}(\log q) - 1 \;>\; \frac{1}{p}\log F(p) + \left(1 - \frac{2}{k+2}\right)(\log q) - 1 \;.$$

Let $N = \lfloor n/(\alpha \log n) \rfloor$, $p = q = N$, and $k = \lfloor \alpha \log n/(1+3\alpha) \rfloor - 1$. We have $n \geqslant (k+1)(p + 3\alpha q)$ as required. So,

$$\begin{aligned} t+1 &\geqslant \frac{1}{N}\log F(N) + \left(1 - \frac{2}{\alpha \log n/(1+3\alpha)}\right)\log N \\ &\geqslant \frac{1}{N}\log F(N) + \log N - \frac{2(1+3\alpha)}{\alpha}\frac{\log N}{\log n} \\ &\geqslant \frac{1}{N}\log F(N) + \log N - 8, \quad \text{as } n \geqslant N \text{ and } \alpha \geqslant 1, \text{ completing the proof.} \end{aligned}$$

□

Let $I(n)$ be the number of labeled interval graphs with n vertices. At every labeled interval graph G with $n-1$ vertices, one can associated a new labeled graph G' obtained by attaching a vertex r, labeled n, to all the vertices of G. The graph G' is an interval 1-graph, choosing $l = r$. Clearly, such interval 1-graphs can be linked to form an interval graph. As the extra vertex is labeled n, all the associated labeled graphs are distinct, and thus their number is at least $I(n-1)$. It turns out that interval graphs have a 1-linkable sub-family with $I(n-1)$ graphs of n vertices.

To get a lower bound on the label length for distance labeling on interval graphs, we can apply Lemma 2.1 with $I(n)$. However, computing $I(n)$ is an unsolved graph-enumeration problem. Cohen, Komlós and Muller gave in [CKM79] the probability $p(n, m)$ that a labeled n-vertex m-edge random graph is an interval graph under conditions on m. They have computed $p(n, m)$ for $m < 4$, and showed that

$$p(n, m) \;=\; \exp\left(-\frac{32c^6}{3}\right) \qquad \text{where } \lim_{n \to +\infty} \frac{m}{n^{5/6}} = c \;.$$

As the total number of labeled n-vertex m-edge graphs is $\binom{\binom{n}{2}}{m}$, it follows a formula of $p(n,m) \cdot \binom{\binom{n}{2}}{m}$ for the number of labeled interval graphs with $m = \Theta(n^{5/6})$ edges. Unfortunately, using this formula it turns out that $I(n) \geqslant 2^{\Omega(n^{5/6}\log n)} = 2^{o(n)}$, a too weak lower bound for our needs. The exact number of interval graphs is given up to 30 vertices in [Han82]. Actually, the generating functions for interval and proper interval graphs (labeled and unlabeled) are known [Han82], but only an asymptotic of $2^{2n+o(n)}$ for unlabeled n-vertex proper interval graphs can be estimated from these equations. In conclusion Hanlon [Han82] left open to know whether the asymptotic on the number of unlabeled interval graphs is $2^{\mathcal{O}(n)}$ or $2^{\Omega(n\log n)}$.

As the number of labeled interval graphs is clearly at least $n! = 2^{(1-o(1))n\log n}$ (just consider a labeled path), the open question of Hanlon is to know whether $I(n) = 2^{(c-o(1))n\log n}$ for some constant $c > 1$. Hereafter we show that $c = 2$, which is optimal.

Theorem 2.5 *The number $I(n)$ of labeled n-vertex connected interval graphs satisfies $\frac{1}{n}\log I(n) \geqslant 2\log n - \log\log n - \mathcal{O}(1)$. It follows that there are $2^{\Omega(n\log n)}$ unlabeled n-vertex interval graphs.*

Proof. As there are at least $I(n)/n!$ unlabeled interval graphs, the lower bound on $I(n)$ implies that $\log(I(n)/n!) \geqslant 2n\log n - n\log n - \mathcal{O}(n) = \Omega(n\log n)$. So there are $2^{\Omega(n\log n)}$ unlabeled interval graphs. Let us prove the lower bound on $I(n)$.

Let $\mathcal{S}_{p,k}$ denote the set of all the sequences $(S_1,\ldots,S_p)$ of integer sets such that, for every $i \in \{1,\ldots,p\}$, it holds: $|S_i| = k$, and $S_i \subset \{0,\ldots,(p-i)(k+1)+k\}$. As there are $\binom{(p-i)(k+1)+k+1}{k}$ ways to choose S_i, it follows that $|\mathcal{S}_{p,k}| = \prod_{i=1}^{p}\binom{(p-i)(k+1)+k+1}{k} = \prod_{i=1}^{p}\binom{i(k+1)}{k}$.

From every $S = (S_1,\ldots,S_p) \in \mathcal{S}_p$ we construct an interval graph G_S of vertex set $V(G_S) = A \cup B \cup C$ with: $A = \{a_{i,j} \mid 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant k\}$, $B = \{b_0,\ldots,b_p\}$, and $C = \{c_1,\ldots,c_k\}$. The edge set is defined by the intersection of the intervals associated to each vertex of G_S. For every $v \in V(G_S)$, the interval of v is $\mathcal{I}(v) \subset [0, p(k+1)+2k]$ and defined as follows: for $i \in \{1,\ldots,p\}$ and $j \in \{1,\ldots,k\}$ (in the following, we denote by $s_{i,j}$ the j-th greatest integer of S_i):

- $\mathcal{I}(a_{i,j}) = [(i-1)(k+1)+j, i(k+1)+s_{i,j}]$;
- $\mathcal{I}(b_i) = [i(k+1), i(k+1)+k]$ and $\mathcal{I}(b_0) = [0,k]$;
- $\mathcal{I}(c_j) = [p(k+1)+j, p(k+1)+j+k]$.

As depicted on Fig. 2 (where $p = k = 3$), for each fixed i, the right boundaries of the $a_{i,j}$'s must differ and must fall in the area delimited by the two vertical dotted lines. The intervals associated with b_i's and c_j's are all of length $k+1$.

We consider the set $\mathcal{F}_{p,k}$ composed of all labeled graphs G_S obtained for $S \in \mathcal{S}_{p,k}$, and such that b_0 is labeled 1 and c_j is labeled $j+1$, for every $j \in \{1,\ldots,k\}$. Clearly, $\mathcal{F}_{p,k}$ is composed of labeled connected interval graphs on $n = (p+1)(k+1)$ vertices. Thus,

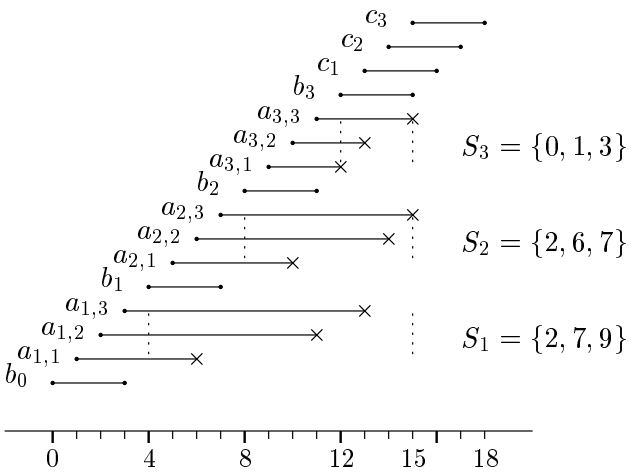

Figure 2: The interval representation of G_S for $S = (S_1, S_2, S_3)$.

$I((p+1)(k+1)) \geqslant |\mathcal{F}_{p,k}|$. We will first show that:

$$\forall\, p, k \geqslant 1, \quad |\mathcal{F}_{p,k}| \;=\; (p(k+1))! \cdot |\mathcal{S}_{p,k}| \tag{1}$$

and then show how to fix $p = p(n)$ and $k = k(n)$ to get the desired lower bound on $I(n)$.

By construction $\mathcal{F}_{p,k}$ contains at most $|\mathcal{S}_{p,k}|$ unlabeled graphs, and each one has at most $(|A| + |B \setminus \{b_0\}|)! = (p(k+1))!$ ways of labeling, since $b_0, c_1, \ldots, c_k$ are labeled $1, \ldots, k+1$. Hence $|\mathcal{F}_{p,k}| \leqslant (p(k+1))!|\mathcal{S}_{p,k}|$. To prove equality, we describe hereafter a 2 step procedure that from any $G \in \mathcal{F}_{p,k}$: 1) retrieves in a unique way a sequence S such that G is isomorphic to G_S; and 2) gives the label sequence of the vertices of G in a given order (for instance in the increasing order of their left boundary in the interval representation of G_S). These two steps ensure that $|\mathcal{F}_{p,k}| \leqslant (|A| + |B \setminus \{b_0\}|)!\, |\mathcal{S}_{p,k}|$.

For that, let us consider any labeled graph $G \in \mathcal{F}_{p,k}$. Let S be a sequence such that G is isomorphic to G_S. (Obviously, at this step, S and the interval representation of G_S are unknown). We say that a vertex v of G is *identified* is one can decide whether $v = a_{i,j}$ or whether $v = b_i$, or whether $v = c_j$ (and then gives for each case the corresponding indices i and j). So identifying all the vertices of G completes Step 2. As we will see later, identifying a vertex v allows also to recover its interval representation and possibly its associated integer $s_{i,j}$, completing Step 1 as well.

For every $i \in \{0, \ldots, p\}$, let L_i denote the set of vertices of G_S whose left boundary is $\leqslant i(k+1)$, i.e., at most equals the left boundary of b_i. To show that all the vertices of G can be identified, it suffices to show that all the vertices of L_p can be identified in G. Indeed, $G_S \setminus L_p$ consists of the vertices of C, and each $c_j \in C$ can be identified in G as its label is $j+1$. So, let us show by induction that all the vertices of L_p can be identified in G.

$L_0 = \{b_0\}$ can be identified in G as its label is 1. Assume that L_{i-1} can be identified, $i > 0$, and let us show how to identify the vertices of L_i. As $L_{i-1} \subset L_i$, we just have to

identify $L_i \setminus L_{i-1} = \{a_{i,1}, \ldots, a_{i,k}\} \cup \{b_i\}$.

Let $G_i = (G_S \setminus L_{i-1}) \cup \{b_{i-1}\}$. In the graph G_i, we have the following properties that comes trivially from the interval definition of G_S:

1. the neighbors of b_{i-1} are $a_{i,1}, \ldots, a_{i,k}$;
2. every vertex of $V(G_i) \setminus \{b_{i-1}\} \setminus C$ is of degree $> k$; and
3. the degree of $a_{i,j}$ is $k + j + s_{i,j}$.

Note that by hypothesis b_{i-1} and L_{i-1} can be identified, so the above properties hold also for the graph $(G \setminus L_{i-1}) \cup \{b_{i-1}\}$. Combining Property 1 and 3, the $a_{i,j}$'s can be identified as the degree of $a_{i,j}$'s are distinct and increasing with j. Moreover, as the graph G_{i+1} can be obtained from G_i by removing the identified vertices $b_{i-1}, a_{i,1}, \ldots, a_{i,k}$, it follows (combining Property 1 and 2 to G_{i+1}), that b_i can be identified: it is the only vertex of $G_{i+1} \setminus C$ that is of degree k exactly. We have showed that all the vertices of L_i can be identified in G. Therefore, all the vertices of L_p, and thus all the vertices of G, can be identified. This completes Step 2.

Recover the sequence S is now easy once $a_{i,j}$'s have been identified in G: it suffices to compute G_i and, by Property 3, to recover $s_{i,j}$ from the degree of $a_{i,j}$. This completes Step 1 of the procedure and thus proves Eq. (1).

Using the fact that $\binom{a}{b} \geqslant (a/b)^b$ we have:

$$|\mathcal{S}_{p,k}| \;\geqslant\; \prod_{i=1}^{p} \left(\frac{i(k+1)}{k}\right)^k = \left(1 + \frac{1}{k}\right)^k (p!)^k \;\geqslant\; (p!)^k \;.$$

Thank to Eq. (1) and using the fact that $x! \geqslant (x/e)^x$, we have:

$$|\mathcal{F}_{p,k}| \;\geqslant\; \left(\frac{p(k+1)}{e}\right)^{p(k+1)} \left(\frac{p}{e}\right)^{pk} \;\geqslant\; \left(\frac{p^2 k}{e^2}\right)^{pk} \;.$$

As $I(n)$ is increasing, for every $n \geqslant (p+1)(k+1)$ we have therefore:

$$\log I(n) \;\geqslant\; \log |\mathcal{F}_{p,k}| \;\geqslant\; 2pk \log p + pk \log k - \mathcal{O}(pk) \;.$$

Let us choose $k = \lfloor \log n \rfloor - 1$ and $p = \lfloor n / \log n \rfloor - 1$ so that $n \geqslant (p+1)(k+1)$. Observe that $pk \geqslant n(1 - 4/\log n)$. We have:

$$\begin{aligned}
\log I(n) &\geqslant 2pk \log(n/\log n) + pk \log\log n - \mathcal{O}(n) \\
&\geqslant 2pk \log n - pk \log\log n - \mathcal{O}(n) \\
&\geqslant 2n(1 - 4/\log n) \log n - n(1 - 4/\log n) \log\log n - \mathcal{O}(n) \\
&\geqslant 2n \log n - n \log\log n - \mathcal{O}(n)
\end{aligned}$$

This ends the proof of Theorem 2.5. □

It is clear that $\frac{1}{n}\log I(n) \leqslant \lceil 2\log(2n)\rceil = 2\log n + \mathcal{O}(1)$, as each vertex can be described by its interval representation (there are at most $2n$ distinct interval boundaries). So the lower bound of Theorem 2.5 on $\frac{1}{n}\log I(n)$ is tight up to an additive factor of $\log\log n$.

Theorem 2.6 *Any distance labeling scheme on the family of n-vertex interval graphs requires a label of length at least $3\log n - 4\log\log n$. In other words (by Proposition 2.1), any universal distance matrix for interval graphs is of dimension $\Omega(n^3/\log^4 n)$.*

Proof. We have seen that the interval graph family has a 1-linkable sub-family with at least $I(n-1)$ graphs of n vertices. By Theorem 2.5 and Lemma 2.1, for $N = \lfloor n/\log n\rfloor$, the length of the maximum label is at least:

$$2\log N - \log\log N + \log N - \mathcal{O}(1) \;\geqslant\; 3\log n - 4\log\log n$$

this completes the proof. □

Theorem 2.7 *Any distance labeling scheme on the family of n-vertex proper interval graphs requires a label of length at least $2\log n - \log\log n - \mathcal{O}(1)$. In other words (by Proposition 2.1), any universal distance matrix for interval graphs is of dimension $\Omega(n^2/\log n)$.*

Proof. Let us construct a large sub-family of proper interval graphs that is 2-linkable, and containing at least $(n-2)!$ graphs of n vertices.

We consider the interval graph G_p with $2p$ vertices defined by: $V(G_p) = \{x_1,\ldots,x_p\} \cup \{y_1,\ldots,y_p\}$, and by the intervals $\mathcal{I}(x_i) = [i, i+p]$ and $\mathcal{I}(y_i) = [i+p, i+2p-1]$, for every $i \in \{1,\ldots,p\}$ (see Fig. 3). The intervals are all of the same length (namely p), so G_p is a proper interval graph.

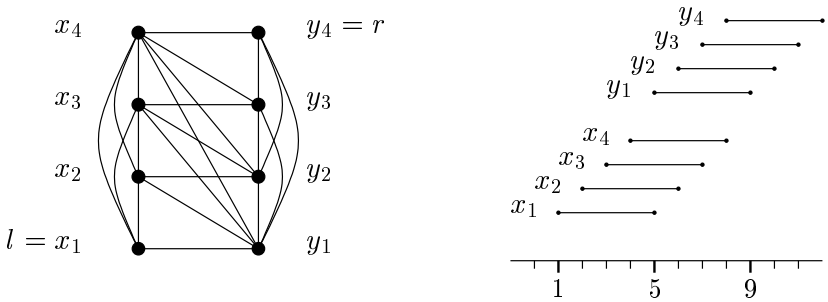

Figure 3: The graph G_4 and its interval representation.

Let consider any labeled graph G isomorphic to G_p such that the vertices x_1 and y_1 are labeled 1 and 2 respectively. It is not difficult to see by induction that once x_i and y_i are identified in G, then x_{i+1} and y_{i+1} can be identified. It suffices to remove all the vertices

$x_1, \ldots, x_{i-1}$ and $y_1, \ldots, y_{i-1}$ from G (and so to form a graph isomorphic to $G_{p-(i-1)}$), and observe that in this latter graph: 1) x_{i+1} is the neighbor of x_i, distinct from y_i, with the minimum degree; and 2) y_{i+1} is the only non-neighbor of x_i that is adjacent to x_{i+1}. Therefore, there are at least $(2p-2)! = (n-2)!$ labeled graphs isomorphic to G_p.

Now, setting $l = x_1$ and $r = y_p$, l and r are both of eccentricity 2, as the vertex x_p is adjacent to all the vertices. So all labeled graphs isomorphic to G_p are 2-graphs. It is easy to see (in particular from the interval representation of G_p) that such 2-graphs can be linked to form a new proper interval graph. So the proper interval graph family has a 2-linkable sub-family with at least $(n-2)!$ graphs of n vertices, as claimed.

We note that $\frac{1}{n}\log(n-2)! \geqslant \log n - \mathcal{O}(1)$. By Theorem 2.5 and Lemma 2.1, for $N = \lfloor n/(2\log n)\rfloor$, the length of the maximum label is at least:

$$\log N + \log N - \mathcal{O}(1) \;\geqslant\; 2\log n - 2\log\log n - \mathcal{O}(1)$$

this completes the proof. □

3 A Scheme for Proper Interval Graphs

Proper interval graphs form a sub-family of interval graphs. A layout of an interval graph is *proper* if there is no intervals strictly contained in another one, i.e., there are no intervals $[a,b]$ and $[c,d]$ with $a < c < d < b$. A proper interval graph is an interval graph having a proper layout.

There are several well known characterizations of this sub-family: G is a proper interval graph iff G is an interval graph without any induced $K_{1,3}$ [Weg67]; G is a proper interval graph iff it has an interval representation using unit length interval [Rob69]. The layout of an interval graph, and a proper layout of a proper interval graph can be computed in $\mathcal{O}(n+m)$ time [CKN+95, dFMdM95, HBJH90], where m is the number of edges.

From now on we consider that the input graph $G = (V, E)$, a connected, unweighted and n-vertex proper interval graph, is given by a proper layout $\mathcal{I}$. If the layout is not proper, we use the general scheme described in Section 4. For every vertex x, we denote by $\textsc{l}(x)$ and $\textsc{r}(x)$ respectively the left and the right boundary of the interval $\mathcal{I}(x)$. As done in [CLSS98], the intervals are assumed to be sorted according to the left boundary, breaking the tie with increasing right boundary. We will also assume that the $2n$ boundaries are pairwise distinct. If not, it is not difficult to see that in $\mathcal{O}(n)$ time one can scan all the boundaries from $\min_x \textsc{l}(x)$ to $\max_x \textsc{r}(x)$ and compute another layout of G that is still proper, and with sorted and distinct boundaries.

Let x_0 be the vertex with minimum right boundary. The *base-path* $[x_0, \ldots, x_k]$ is the sequence of vertices such that $\forall i > 0$, x_i is the neighbor of x_{i-1} whose $\textsc{r}(x_i)$ is maximum.

The *layer partition* $V_1, \ldots, V_k$ is defined by:

$$V_i = \{v \mid \text{L}(v) < \text{R}(x_{i-1})\} \setminus \bigcup_{0 \leqslant j < i} V_j \quad \text{with } V_0 = \varnothing.$$

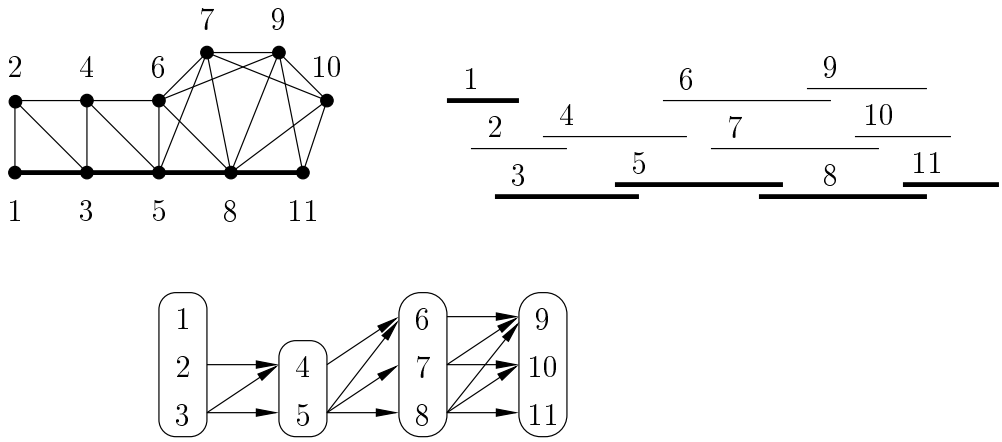

Figure 4: A proper interval graph with an interval representation and the associated layer partition. The base-path is in bold.

Given the sorted list of intervals, the base-path and the layer partition can be computed in $\mathcal{O}(n)$ time. Let $\lambda(x)$ denote the unique index i such that $x \in V_i$. Let H be the digraph on the vertex set V composed of all the arcs xy such that $\lambda(x) < \lambda(y)$ and $(x, y) \in E$. Note that H is a directed acyclic graph. Let H^t denote the transitive closure of H (since H is acyclic, H^t defines a poset), and let $\text{adj}_{H^t}(x, y)$ be the boolean function such that:

$$\text{adj}_{H^t}(x, y) = 1 \text{ iff } xy \text{ is an arc of } H^t \ .$$

The scheme we proposed is based on the following theorem:

Theorem 3.1 *For all distinct vertices x and y such that $\lambda(x) \leqslant \lambda(y)$:*

$$\text{dist}_G(x, y) = \lambda(y) - \lambda(x) + 1 - \text{adj}_{H^t}(x, y) \tag{2}$$

To prove it, we need of this preliminary result.

Lemma 3.1 *$\{x_{i-1}\} \cup V_i$ induced a clique in G, for every $i \geqslant 1$.*

Proof. Let us first show that x_{i-1} dominates V_i. By definition of x_0, any vertex $v \in V_1$ verifies $\text{L}(v) < \text{R}(x_0) < \text{R}(v)$, therefore V_1 is dominated by x_0. Let v be a vertex of V_i with $i > 1$. By definition of the layer partition, $\text{L}(v) < \text{R}(x_{i-1})$ and since $v \notin V_j$ for $j < i$, we also have $\text{R}(x_{i-2}) < \text{L}(v)$. Moreover x_{i-2} is adjacent to x_{i-1} therefore $\text{L}(x_{i-1}) < \text{R}(x_{i-2}) < \text{L}(v) < \text{R}(x_{i-1})$ and x_{i-1} dominates V_i.

Let us show now that V_i induces a clique. By definition of x_0, any vertex $v \in V_1$ verifies $\text{L}(v) < \text{R}(x_0) < \text{R}(v)$ therefore the intervals of vertices of V_1 pairwise intersect and V_1 is a clique. Let us consider V_i with $i > 2$ and assume that there are two non-adjacent vertices u and v in V_i. As noticed before, for any vertex $w \in V_i$, we have $\text{L}(x_{i-1}) < \text{R}(x_{i-2}) < \text{L}(w) < \text{R}(x_{i-1})$. W.l.o.g. assume that $\text{L}(u) < \text{R}(u) < \text{L}(v) < \text{R}(v)$. Therefore $\text{L}(x_{i-1}) < \text{R}(x_{i-2}) < \text{L}(u) < \text{R}(u) < \text{L}(v) < \text{R}(x_{i-1})$: contradiction the set of vertices $\{x_{i-2}, x_{i-1}, u, v\}$ induces a $K_{1,3}$. □

Proof of Theorem 3.1. Let x and y be two vertices such that $\lambda(x) = i$ and $\lambda(y) = i + h$ with $h \geqslant 0$. Note that the theorem is true for $h = 0$ since V_i is a clique (by Lemma 3.1) and for $h = 1$ since x_i neighbors y and x (by Lemma 3.1).

So assume that $h > 1$. First let us show that any u, v such that $\lambda(v) - \lambda(u) > 1$ are non-adjacent. Notice that definition of the layer partition implies $\text{R}(x_i) < \text{L}(v)$. Since $\text{R}(u) \leqslant \text{R}(x_i)$, v cannot neighbor u. It follows that $\lambda(y) - \lambda(x) \leqslant \text{dist}_G(x, y)$ and if equality holds then by definition $\text{adj}_{H^t}(x, y) = 1$.

Let us consider the sub-path $[x_i, \ldots, x_{i+h-1}]$ of the base-path. By Lemma 3.1, x_{i+h-1} is adjacent to y. By Lemma 3.1, V_i is a clique, therefore x is adjacent to x_i. So we are able to construct a path of length $h+1$ between x and y, namely $P = [x, x_i, \ldots, x_{i+h-1}, y]$: $\text{dist}_G(x, y) \leqslant \lambda(y) - \lambda(x) + 1$. Finally if equality holds, it means that x does not neighbor any vertex $z \in V_{i+1}$ such that $\text{adj}_{H^t}(z, y) = 1$, therefore $\text{adj}_{H^t}(x, y) = 0$. □

Therefore to compute the distance between any pair of vertices we have to test whether $\text{adj}_{H^t}(x, y) = 1$. For this reason H^t can be considered as the *graph of errors* associated to the layer partition.

Theorem 3.2 *There exists a linear ordering π of the vertices, constructible in $\mathcal{O}(n)$ time, such that:*

$$\text{adj}_{H^t}(x, y) = 1 \text{ iff } \lambda(x) < \lambda(y) \text{ and } \pi(y) < \pi(x) \tag{3}$$

Proof. The ordering π is defined by the following simple algorithm:

π is the pop ordering of a DFS on H using $\text{L}(x)$ as a priority rule.

This algorithm can be seen as an elimination ordering of the vertices such that at each step, the sink of the subgraph of H induced by the non-eliminated vertices, that has a minimum left bound, is removed. On the example of Fig. 4, we obtain the following linear ordering:

$$\pi \;=\; 1, 9, 6, 4, 2, 10, 7, 11, 8, 5, 3$$

This algorithm can be implemented to run in $\mathcal{O}(n)$ time. First the digraph H can be stored in $\mathcal{O}(n)$ space. The set of vertices are sorted in an array with respect to the left boundaries of their interval. Notice that the layers and also the neighborhood in H of any

vertex appear consecutively in this array. So each vertex is associated to its first and last vertex in the array.

The priority rule implies that when a vertex v is popped by the algorithm, any vertex u of the same layer such that $\text{L}(u) < \text{L}(v)$ has already been popped. It means that when a vertex is at the top of the stack during the DFS, we can find in $\mathcal{O}(1)$ the next vertex to be pushed if the layer of each vertex is also stored and for each layer, the index of the last popped vertex is maintained.

Let us now prove the correctness of Eq. (3).

$\Rightarrow$ It is easy to check that if there is an arc from x to y in H^t, then $\lambda(x) < \lambda(y)$ and $\pi(y) < \pi(x)$.

$\Leftarrow$ So let x and y be two non-adjacent vertices in H^t such that $\lambda(x) < \lambda(y)$ (it implies that $\text{L}(x) < \text{L}(y)$).

We first show that $\forall z$ such that $xz \in E(H^t)$, then $yz \in E(H^t)$. If $\lambda(x) = \lambda(y)$, yz exists since $\text{L}(x) < \text{L}(y)$. So assume that $\lambda(x) < \lambda(y)$. If xz exists but not yz, then we can enlighten a $K_{1,3}$ containing x, y and z with a vertex $v \in V_{\lambda(z)}$ that belongs to a shortest x, z-path in G: contradiction, G is a proper interval graph.

By the way at the step y becomes a sink, there remains no vertex z such that $\lambda(y) < \lambda(z)$ and $xz \in E(H^t)$. If x is also a sink, then by the priority rule $\pi(x) < \pi(y)$. If it is not the case, then $\lambda(x) < \lambda(y)$ and there exists a sink x' such that $xx' \in E(H^t)$ and $\lambda(x') \leqslant \lambda(y)$. Notice that $\text{L}(x') < \text{L}(y)$. Otherwise we would have $\lambda(x') = \lambda(y)$ and the existence of a shortest x, x'-path of length $\lambda(x') - \lambda(x)$ (see Theorem 3.1), would implies the existence of a x, y-path of same length. Therefore xy would be an arc of H^t: contradiction. Since $\text{L}(x') < \text{L}(y)$, the priority rule implies that x' is popped before y. Recursively applying this argument, we can prove that x will be popped before y by the algorithm and so $\pi(y) < \pi(x)$.

□

The *dimension* of a poset P is the minimum number d of linear orderings $\rho_1, \ldots, \rho_d$ such that $x <_P y$ iff $x <_{\rho_i} y$ for every i. The next corollary follows immediately from Theorem 3.2.

Corollary 3.3 *The graph of errors H^t is a poset of dimension two.*

To each vertex x of G, we assign the label $L(x, G) = \langle \lambda(x), \pi(x) \rangle$. The distance decoder is given by Eq. (2) and (3). Clearly, $\lambda(x), \pi(x) \in \{1, \ldots, n\}$, so:

Theorem 3.4 *The family of n-vertex proper interval graphs enjoys a distance labeling scheme using labels of length $2\lceil \log n \rceil$, and the distance decoder has constant time complexity. Moreover, given the sorted list of intervals, all the labels can be computed in $\mathcal{O}(n)$ time.*

From Proposition 2.1,

Corollary 3.5 *There is an universal distance matrix of dimension $\mathcal{O}(n^2)$ for n-vertex proper interval graphs.*

Actually, the bound of Corollary 3.5 is close to optimality. We can show a lower bound of $\Omega((n/\log n)^2)$. The proof, based on an extension of Lemma 2.1 and on an enumeration of proper interval graphs of diameter two, will appear in the final version.

4 A Scheme for Interval Graphs

In this section, we assume that $G = (V, E)$ is an interval graph, connected and unweighted, and that a layout $\mathcal{I}$ of G is given by a sorted list of intervals. Up to an $\mathcal{O}(n)$ preprocessing time, we assume that interval boundaries are pairwise distinct and taken from the set $\{1, \ldots, 2n\}$.

From $\mathcal{I}$ we will build a family of intervals $\mathcal{I}'$ that is the layout of a proper interval graph $G' = (V, E')$ so that the distance in G can be retrieved from the distances in G' plus some additional information. From now $\textsc{l}(x)$ and $\textsc{r}(x)$ will denote the boundaries of $\mathcal{I}(x)$, and $\textsc{l}'(x)$ and $\textsc{r}'(x)$ the boundaries of $\mathcal{I}'(x)$.

Let x be a vertex of V such that $\mathcal{I}(x) \subset \mathcal{I}(y)$ for some vertex y, then the *enclosed neighbor* of x, denoted by $\mathrm{N_e}(x)$, is the neighbor of x such that $\mathcal{I}(x) \subset \mathcal{I}(\mathrm{N_e}(x))$ and such that $\textsc{r}(\mathrm{N_e}(x))$ is maximum. As the boundaries of $\mathcal{I}$ are pairwise disjoint, the enclosed neighbor, if it exists, is unique for every vertex. The layout $\mathcal{I}'$ is obtained from $\mathcal{I}$ as follows: if x has an enclosed neighbor then we set $\mathcal{I}'(x) = [\textsc{l}(x), \textsc{r}(\mathrm{N_e}(x))]$, and we set $\mathcal{I}'(x) = \mathcal{I}(x)$ otherwise. Let us remark: 1) $\textsc{l}'(x) = \textsc{l}(x)$ and $\textsc{r}'(x) \geqslant \textsc{r}(x)$; 2) by maximality of its right boundary $\mathcal{I}(\mathrm{N_e}(x)) = \mathcal{I}'(\mathrm{N_e}(x))$; and 3) G is a subgraph of G' and thus $E \subseteq E'$.

Lemma 4.1 *The layout $\mathcal{I}'$ of G' is proper.*

Proof. Assume there is a pair of vertices u and v such that $\mathcal{I}'(u) \subset \mathcal{I}'(v)$ and $\textsc{r}'(u) < \textsc{r}'(v)$. W.l.o.g. that assume v is *maximal*, i.e., $\textsc{r}'(v)$ is maximum and, if equality, we choose the one with minimum $\textsc{l}'(v)$.

As $\textsc{r}(u) \leqslant \textsc{r}'(u)$, we have therefore $\textsc{l}(v) < \textsc{l}(u) < \textsc{r}(u) < \textsc{r}'(v)$. If $\textsc{r}(v) = \textsc{r}'(v)$, then u has an enclosed neighbor and thus $\textsc{r}'(u) \geqslant \textsc{r}(v) = \textsc{r}'(v)$: a contradiction with $\textsc{r}'(u) < \textsc{r}'(v)$. So $\textsc{r}(v) < \textsc{r}'(v)$. It follows that v has an enclosed neighbor w. $\mathcal{I}(v) \subset \mathcal{I}(w) = \mathcal{I}'(w)$. Thus, $\textsc{l}(w) < \textsc{l}(v) < \textsc{r}'(v) \leqslant \textsc{r}'(w)$. Thus v is not maximal: a contradiction. □

Note that in the above transformation, the proper layout obtained may have $\mathcal{I}'(x) \subset \mathcal{I}'(y)$ but with $\textsc{r}'(x) = \textsc{r}'(y)$. However $\mathcal{I}'$ is a proper layout, so the scheme of Section 3 applies. We say that a vertex x has been *extended* in G' if $\textsc{r}'(x) > \textsc{r}(x)$.

Lemma 4.2 *If x and y are two vertices such that $\mathrm{R}(x) = \mathrm{R}'(x) \leqslant \mathrm{R}(y)$. Then $\mathrm{dist}_G(x,y) = \mathrm{dist}_{G'}(x,y)$.*

Proof. Let us first remark that since G is a subgraph of G', then $\mathrm{dist}_G(x,y) \geqslant \mathrm{dist}_{G'}(x,y)$. So assume that $\mathrm{dist}_G(x,y) = k > 1$. Let us prove there exists a shortest x,y-path in G' that uses no extended vertices and that is also a shortest path in G. Let $P' = [x = u_0, u_1, \ldots, u_j, \ldots, u_k = y]$ be a shortest x,y-path in G' such that u_j is extended ($\mathcal{I}(u_j) \neq \mathcal{I}'(u_j)$). Since $\mathcal{I}(x) = \mathcal{I}'(x)$, x and u_1 are adjacent in G. Notice that $\mathrm{R}(x) \leqslant \mathrm{R}(y)$ implies that for any i, $0 < i \leqslant k$, $\mathrm{R}(u_{i-1}) \leqslant \mathrm{R}(u_i)$. Since $\mathcal{I}(u_j) \subset \mathcal{I}(\mathrm{N_e}(u_j))$, vertex u_j can be substituted in P' by $\mathrm{N_e}(u_j)$. Since $\mathrm{N_e}(u_j)$ is not extended, any extended vertex of P' can be substituted by its enclosed neighbor. □

For any vertex x, define the *maximum neighbor* of x, denoted by $\mathrm{N_m}(x)$, as the neighbor of x such that $\mathrm{R}(\mathrm{N_m}(x))$ is maximum. Like enclosed neighbors, maximum neighbors are not extended in G'. Otherwise we would have $\mathcal{I}(\mathrm{N_m}(x)) \subset \mathcal{I}(v)$ for some vertex v and $\mathrm{R}(\mathrm{N_m}(x)) < \mathrm{R}(v)$. Since $\mathcal{I}(x) \cap \mathcal{I}(\mathrm{N_m}(x)) \neq \varnothing$, we also have $\mathcal{I}(x) \cap \mathcal{I}(v) \neq \varnothing$. Since $\mathrm{R}(\mathrm{N_m}(x)) < \mathrm{R}(v)$, by definition, we should have $v = \mathrm{N_m}(x)$: contradiction.

Theorem 4.1 *Let x and y be two vertices such that $\mathrm{R}(x) \leqslant \mathrm{R}(y)$. Then,*

$$\mathrm{dist}_G(x,y) = \mathrm{dist}_{G'}(\mathrm{N_m}(x), y) + 1 - \mathrm{adj}_G(x,y)\ . \tag{4}$$

Proof. If x and y are adjacent, then $\mathrm{N_m}(x)$ and y are also adjacent in both G and G', therefore the result holds. So assume that x and y are not adjacent. Since $\mathrm{N_m}(x)$ has not been extended in G', by Lemma 4.2, $\mathrm{dist}_{G'}(\mathrm{N_m}(x), y) = \mathrm{dist}_G(\mathrm{N_m}(x), y)$. So we just have to prove that $\mathrm{dist}_G(x,y) = 1 + \mathrm{dist}_G(\mathrm{N_m}(x), y)$.

Let us first remark that $\mathrm{dist}_G(\mathrm{N_m}(x), y) \leqslant \mathrm{dist}_G(x,y)$ since $\mathrm{R}(x) < \mathrm{L}(y)$ and $\mathrm{R}(x) < \mathrm{R}(\mathrm{N_m}(x))$. We just have to consider the case $\mathrm{dist}_G(\mathrm{N_m}(x), y) = k > 1$. Let $P = [\mathrm{N_m}(x), v_1, \ldots, v_{k-1}, y]$ be a shortest $\mathrm{N_m}(x), y$-path between. Let us assume that $\mathrm{dist}_G(x,y) = k$ then there exists a path $Q = [x, u_1, u_2, \ldots, u_{k-1}, y]$ of length k. By definition of $\mathrm{N_m}(x)$, we have $\mathrm{R}(u_1) < \mathrm{R}(\mathrm{N_m}(x))$. It implies that $\mathrm{N_m}(x)$ is adjacent to u_2, therefore $[\mathrm{N_m}(x), u_2, \ldots, u_{k-1}, y]$ is a path of length $k-1$: contradiction. □

Eq. (4) directly gives a distance labeling scheme for the family of interval graphs. To each vertex, we have to associate $L(x,G)$:

- $\mathrm{R}(x)$ and $\mathrm{L}(x)$ the boundaries of $\mathcal{I}(x)$;
- $L(x,G')$ and $L(\mathrm{N_m}(x), G')$, the labels of x and $\mathrm{N_m}(x)$ in the proper interval graph G'.

The distance decoder is the function described in Eq. (4). It uses the distance decoder of proper interval graph given in the previous section. These labels, composed a priori of 6 integers, can be compacted. Indeed, there is no possible edge in G between any pair of

vertices that do not belong to the same or consecutive layers of the layer partition. Thus, we have either $\lambda(\mathrm{N_m}(x)) = \lambda(x)$ or $\lambda(\mathrm{N_m}(x)) = \lambda(x) + 1$. We do not need to store the pair $\langle \lambda(x), \lambda(\mathrm{N_m}(x)) \rangle$. A single bit, say $b(x)$, suffices to recover the second part of this information. Therefore the label of a vertex x is:

$$L(x, G) = \langle \mathrm{L}(x), \mathrm{R}(x), \lambda(x), \pi(x), b(x), \pi(\mathrm{N_m}(x)) \rangle$$

As $1 \leqslant \mathrm{L}(x) < \mathrm{R}(x) \leqslant 2n$, so $2\lceil \log n \rceil + 2$ bits suffice two store the two first integers. The 4 next components can be stored with $3\lceil \log n \rceil + 1$ bits, summing up to $5\lceil \log n \rceil + 3$. Since the distance labels in proper interval graphs can be computed in $\mathcal{O}(n)$, and since G' can be obtained within the same complexity, all the labels of G can be computed in $\mathcal{O}(n)$ time.

Theorem 4.2 *The family of n-vertex interval graphs enjoys a distance labeling scheme using labels of length $5\lceil \log n \rceil + 3$, and the distance decoder has constant time complexity. Moreover, given the sorted list of intervals, all the labels can be computed in $\mathcal{O}(n)$ time.*

From Proposition 2.1,

Corollary 4.3 *There is an universal distance matrix of dimension $\mathcal{O}(n^5)$ for n-vertex interval graphs.*

5 A Scheme for Circular-Arc Graphs

Circular-arc graphs are a natural generalization of interval graphs, in which vertices correspond to arcs of a circle rather than intervals on the real line. Consider an circular-arc graph G. We can associate to every vertex x of G a range $I(x) = [\theta_r(x), \theta_l(x)]$ of angles where $\theta_r(x)$ and $\theta_l(x)$ are taken clockwise in $[0, 2\pi)$. The range $I(x)$ can be seen as a "cyclic" interval in which $[\theta_r(x), \theta_l(x)]$ for $\theta_r(x) > \theta_l(x)$ stands for $[\theta_r(x), 2\pi) \cup [0, \theta_l(x)]$. The vertices x and y are adjacent if $I(x) \cap I(y) \neq \varnothing$. See an example on Fig. 5.

A neighbor y of x is a *right neighbor* if $(\theta_r(y) - \theta_r(x) \bmod 2\pi) \geqslant 0$. A path $x_0, x_1, \ldots, x_k$ is a *right-path* from x_0 to x_k if, for every $i \in \{0, \ldots, k-1\}$, x_{i+1} is a right neighbor of x_i. Intuitively, a right-path is obtained starting from x_0 and successively clockwise encountering the vertices $x_1, \ldots, x_k$.

Lemma 5.1 *Every shortest path between x and y is either a right-path from x to y, or a right-path from y to x.*

Proof. Assume that $x = x_0, x_1, \ldots, x_k = y$ is a shortest path from x to y but is neither a right-path from x_0 to x_k nor a right-path from x_k to x_0.

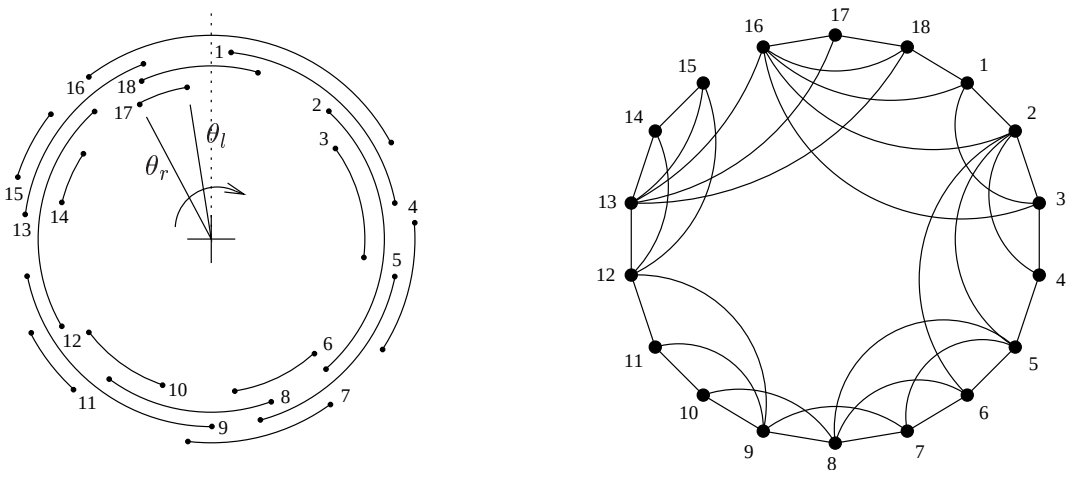

Figure 5: A circular-arc graph with its arc representation. Vertices are labeled from 1 to 18 according to their value θ_r.

W.l.o.g. assume that x_1 is a right neighbor of x_0. The case x_1 is not a right neighbor of x_0 can be treated similarly by exchanging the role of θ_r and θ_l boundaries. Since the path from x_0 to x_k is not right path so there is an integer i such that x_{i+1} is not a right neighbor of x_i and such that the sub-path from x_0 to x_i is a right-path, $1 \leqslant i < k$. In particular, $\theta_r(x_{i-1}) \leqslant \theta_r(x_i)$ and $\theta_r(x_i) > \theta_r(x_{i+1})$. Because x_{i+1} neighbors x_i we have $\theta_l(x_{i+1}) \geqslant \theta_r(x_i)$, and thus $\theta_r(x_i) \in I(x_{i+1})$. Similarly, x_{i-1} neighbors x_i thus $\theta_l(x_{i-1}) \geqslant \theta_r(x_i)$, and thus $\theta_r(x_i) \in I(x_{i-1})$. It follows that $\theta_r(x_i) \in I(x_{i-1}) \cap I(x_{i+1})$, and hence that x_{i-1} neighbors x_{i+1}: a contradiction with the fact that $x_0, \ldots, x_{i-1}, x_i, x_{i+1}$ is a shortest path. □

Let $x_1, \ldots, x_n$ the vertices of G ordered such that $\theta_r(x_i) \leqslant \theta_r(x_{i+1})$ for every $i \in \{1, \ldots, n-1\}$. We associate to G a new intersection graph $\widetilde{G}$ with vertex set $V(\widetilde{G}) = \bigcup_{1 \leqslant i \leqslant n} \{I(x_i^1), I(x_i^2)\}$ where, for every $1 \leqslant i \leqslant n$:

$$I(x_i^1) = \begin{cases} [\theta_r(x_i), \theta_l(x_i)] & \text{if } \theta_r(x_i) \leqslant \theta_l(x_i) \\ [\theta_r(x_i), \theta_l(x_i) + 2\pi] & \text{if } \theta_r(x_i) > \theta_l(x_i) \end{cases}$$

$$I(x_i^2) = \begin{cases} [\theta_r(x_i) + 2\pi, \theta_l(x_i) + 2\pi] & \text{if } \theta_r(x_i) \leqslant \theta_l(x_i) \\ [\theta_r(x_i) + 2\pi, \theta_l(x_i) + 4\pi] & \text{if } \theta_r(x_i) > \theta_l(x_i) \end{cases}$$

Intuitively, $\widetilde{G}$ is obtained from G by listing all the intervals according to increasing angle θ_r, starting from the arc of x_1, and clockwise turning two rounds. So, each vertex x_i in G appears twice in $\widetilde{G}$ as x_i^1 and x_i^2. We check that $\widetilde{G}$ is an interval graph, and that if G is a proper circular-arc graph (i.e., there is no two vertices x, y with $I(x) = [a, b]$, $I(y) = [c, d]$ and $a < c < d < b$), then $\widetilde{G}$ is a proper interval graph.

Lemma 5.2 *For every $i < j$,* $\mathrm{dist}_G(x_i, x_j) = \min\left\{\mathrm{dist}_{\widetilde{G}}(x_i^1, x_j^1), \mathrm{dist}_{\widetilde{G}}(x_j^1, x_i^2)\right\}$.

Proof. Let $P = [x_i, x_{t_1}, \ldots, x_{t_k}, x_j]$ be a shortest path between x_i and x_j in G. From Lemma 5.1 P is a right-path from x_i to x_j, or a right-path from x_j to x_i.

In the former case, we have $i \leqslant t_1 \leqslant \cdots \leqslant t_k \leqslant j$. Consider the path $P' = [x_i^1, x_{t_1}^1, \ldots, x_{t_k}^1, x_j^1]$ of $\widetilde{G}$. Because P' can be constructed from P by turning clockwise from x_i to x_j in G, P' is isomorphic to P, showing that $\mathrm{dist}_G(x_i, x_j) \leqslant \mathrm{dist}_{\widetilde{G}}(x_i^1, x_j^1)$. We observe that any shortest path from x_i^1 to x_j^1 in $\widetilde{G}$ must be a right-path. By construction, every right-path of length l between a given pair of vertices in $\widetilde{G}$ is isomorphic to some right-path of same length between the corresponding vertices in G. So a shorter path between x_i^1 and x_j^1 would provide a shorter path between x_i and x_j in G. Therefore, P' is a shortest path in $\widetilde{G}$ and $\mathrm{dist}_G(x_i, x_j) = \mathrm{dist}_{\widetilde{G}}(x_i^1, x_j^1)$.

Assume now that P is a right-path from x_j to x_i. P is obtained by turning clockwise from x_j to x_i. P is isomorphic to a right-path P'' in $\widetilde{G}$ between x_j^1 and x_i^2. Indeed, since $i < j$, $\theta_r(x_i) \leqslant \theta_r(x_j) < \theta_r(x_i) + 2\pi$. Similarly to the previous case, P'' is a shortest path, proving that, in this case, $\mathrm{dist}_G(x_i, x_j) = \mathrm{dist}_{\widetilde{G}}(x_j^1, x_i^2)$ and completing the proof of the lemma. □

Therefore, a distance labeling scheme for interval graphs can be transformed into a scheme for circular-arc graph family by doubling the number of vertices, and the label length.

Theorem 5.1 *The family of n-vertex circular-arc graphs enjoys a distance labeling scheme using labels of length $\mathcal{O}(\log n)$, and the distance decoder has constant time complexity. Moreover, given the sorted list of intervals, all the labels can be computed in $\mathcal{O}(n)$ time. So, n-vertex circular-arc graphs has an universal distance matrix of dimension polynomial in n.*

References

[ABF+96] R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). In 7^{th} *Symposium on Discrete Algorithms (SODA)*, pages 365–372. ACM-SIAM, January 1996.

[ABR03] S. Alstrup, P. Bille, and T. Rauhe. Labeling schemes for small distances in trees. In 15^{th} *Symposium on Discrete Algorithms (SODA)*. ACM-SIAM, January 2003.

[AGKR02] S. Alstrup, C. Gavoille, H. Kaplan, and T. Rauhe. Nearest common ancestors: A survey and a new distributed algorithm. In 14^{th} *Annual ACM Symposium on Parallel Algorithms and Architecture (SPAA)*, pages 258–264. ACM PRESS, August 2002.

[AKM01] S. Abitboul, H. Kaplan, and T. Milo. Compact labeling schemes for ancestor queries. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 547–556, 2001.

[AR02a] S. Alstrup and T. Rauhe. Improved labeling scheme for ancestor queries. In 13^{th} *Symposium on Discrete Algorithms (SODA)*, pages 947–953. ACM-SIAM, January 2002.

[AR02b] S. Alstrup and T. Rauhe. Small induced-universal graphs and compact implicit graph representations. In 43^{rd} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, November 2002.

[BV94] O. Berkman and U. Vishkin. Finding level-ancestors in trees. *Journal of Computer and System Sciences*, 48(2):214–230, April 1994.

[Chu90] F.R.K. Chung. Universal graphs and induced-universal graphs. *Journal of Graph Theory*, 14(4):443–454, 1990.

[CKM79] J.E. Cohen, J. Komlós, and T. Mueller. The probability of an interval graph and why it matters. In *Symposia in Pure Mathematics*, volume 34, pages 97–115, 1979.

[CKN+95] D.G. Corneil, H. Kim, S. Natarajan, S. Olariu, and A.P. Sprague. Simple linear time algorithm of unit interval graphs. *Information Processing Letters*, 55(2):99–104, 1995.

[CLSS98] D.Z. Chen, D.T. Lee, R. Sridhar, and C.N. Sekharan. Solving the all-pair shortest path query problem on interval and circular-arc graphs. *Networks*, 31(4):249–257, 1998.

[CV01] B. Courcelle and R. Vanicat. Query efficient implementation of graphs of bounded clique width. *Discrete Applied Mathematics*, 2001. To appear.

[dFMdM95] C.M. Herrera de Figueiredo, J. Meidanis, and C. Picinin de Mello. A linear-time algorithm for proper interval recognition. *Information Processing Letters*, 56(3):179–184, 1995.

[Gav01] C. Gavoille. Routing in distributed networks: Overview and open problems. *ACM SIGACT News - Distributed Computing Column*, 32(1):36–52, March 2001.

[GKK+01] C. Gavoille, M. Katz, N.A. Katz, C. Paul, and D. Peleg. Approximate distance labeling schemes. In 9^{th} *Annual European Symposium on Algorithms (ESA)*, volume 2161 of Lecture Notes in Computer Science, pages 476–488. Springer, August 2001.

[GP01a] C. Gavoille and C. Paul. Split decomposition and distance labelling: An optimal scheme for distance hereditary graphs. In *European Conference on Combinatorics, Graph Theory and Applications (COMB)*, volume 10. Electronical Notes in Discrete Mathematics, September 2001.

[GP01b] C. Gavoille and D. Peleg. Compact and localized distributed data structures. Research Report RR-1261-01, LaBRI, University of Bordeaux, 351, cours de la Libération, 33405 Talence Cedex, France, August 2001. To appear in Journal of Distributed Computing for the PODC 20-Year Special Issue.

[GPPR01] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance labeling in graphs. In 12^{th} *Symposium on Discrete Algorithms (SODA)*, pages 210–219. ACM-SIAM, January 2001.

[Han82] P. Hanlon. Counting interval graphs. *Transactions of the American Mathematical Society*, 272:383–426, 1982.

[HBJH90] P. Hell, J. Bang-Jensen, and J. Huang. Local tournaments and proper circular arc graphs. In *Algorithms, International Symposium SIGAL*, volume 450 of *Lecture Notes in Computer Science*, pages 101–108, Tokyo, 1990.

[KKP00] M. Katz, N.A. Katz, and D. Peleg. Distance labeling schemes for well-separated graph classes. In 17^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1770 of Lecture Notes in Computer Science, pages 516–528. Springer Verlag, February 2000.

[KKP02] M. Katz, N.A. Katz A. Korman, and D. Peleg. Labeling schemes for flow and connectivity. In 13^{th} *Symposium on Discrete Algorithms (SODA)*, pages 927–936. ACM-SIAM, January 2002.

[KMS02] H. Kaplan, T. Milo, and R. Shabo. A comparison of labeling schemes for ancestor queries. In 14^{th} *Symposium on Discrete Algorithms (SODA)*. ACM-SIAM, January 2002.

[KNR92] S. Kannan, M. Naor, and S. Rudich. Implicit representation of graphs. *SIAM Journal of Discrete Mathematics*, 5(4):596–603, 1992.

[KPR02] A. Korman, D. Peleg, and Y. Rodeh. Labeling schemes for dynamic tree networks. In 19^{th} *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2285 of Lecture Notes in Computer Science, pages 76–87. Springer, March 2002.

[Pel00a] D. Peleg. Informative labeling schemes for graphs. In 25^{th} *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 1893 of Lecture Notes in Computer Science, pages 579–588. Springer, August 2000.

[Pel00b] D. Peleg. Proximity-preserving labeling schemes. *Journal of Graph Theory*, 33:167–176, 2000.

[Rob69] F.S. Roberts. Indifference graphs. In *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, 1969.

[Spi02] J.P. Spinrad. Efficient graph representations, 2002. In preparation.

[Tho01] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. In 42^{th} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, October 2001.

[TZ01a] M. Thorup and U. Zwick. Approximate distance oracles. In 33^{rd} *Annual ACM Symposium on Theory of Computing (STOC)*, pages 183–192, Hersonissos, Crete, Greece, July 2001.

[TZ01b] M. Thorup and U. Zwick. Compact routing schemes. In 13^{th} *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 1–10, Hersonissos, Crete, Greece, July 2001. ACM PRESS.

[Weg67] G. Wegner. *Eigenschaften der Neven homologish-einfacher Familien im* R^n. PhD thesis, University of Göttingen, 1967.