

Optimal distance labeling for interval graphs and related graphs families*

Cyril Gavoille[†]

Christophe Paul[‡]

July 2, 2005

Abstract

A distance labeling scheme is a distributed graph representation that assigns labels to the vertices and enables to answer distance queries between any pair (x, y) of vertices by using only the labels of x and y . This paper presents an optimal distance labeling scheme with labels of $\mathcal{O}(\log n)$ bits for the n -vertex interval graphs family. It improves by $\log n$ factor the best known upper bound of [KKP00]. Moreover, the scheme supports constant time distance queries, and if the interval representation of the input graph is given, then the set of labels can be computed in $\mathcal{O}(n)$ time. Our result is tight as we show that the length of any label is at least $3 \log n - \mathcal{O}(\log \log n)$ bits. This lower bound derives from a new estimator of the number of unlabeled n -vertex interval graphs, that is $2^{\Omega(n \log n)}$. To our knowledge, interval graphs are thereby the first known hereditary family with $2^{\Omega(n f(n))}$ unlabeled elements and with a distance labeling scheme with $f(n)$ bit labels.

Keywords: graph representation, distributed data-structure, distance.

1 Introduction

From an efficient graph representation, we could expect a small amount of memory space and fast elementary routines to answer queries like adjacency (see e.g. [Spi03] for a recent overview). Among the classical type of queries, let us mention those arising from the field of communication network and distributed computing, like routing, connectivity and distance queries. Peleg [Pel00a] introduced the concept of *informative labeling*

*An extended abstract of this work appears in *Algorithms - ESA 2003*, proceedings of 11th Annual European Symposium on Algorithms, Lecture Notes in Computer Science, volume 2832

[†]Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I, France. gavoille@labri.fr

[‡]CNRS - Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier, France. paul@lirmm.fr

schemes which generalizes the *implicit graph representation* proposed in [KNR92]. Like classical graph representations, informative labeling schemes have the property of being a distributed data-structure: labels are assigned to the vertices and queries can be answered using only the labels of the involved vertices. Schemes providing compact labels play an important role for localized distributed data-structures (see [GP03b] for a survey).

This paper focusses on *distance labeling schemes* [Pel00b] which, for a graph family \mathcal{F} , is a pair $\langle L, f \rangle$ of functions such that for any graph $G = (V, E) \in \mathcal{F}$:

- $L(v, G)$ is a binary label associated to the vertex v in the graph G , and
- $\forall x, y \in V, f(L(x, G), L(y, G)) = \text{dist}_G(x, y)$, the distance in G between x and y .

A scheme is an $\ell(n)$ -distance labeling if for every n -vertex graph $G \in \mathcal{F}$, the length of the labels is at most $\ell(n)$ bits. The first results on distance labeling scheme [GP01] show that the family of n -vertex graphs requires $\Theta(n)$ bit labels while n -vertex trees only need $\Theta(\log^2 n)$ bit labels¹. The case of dynamic tree networks has been studied in [KPR02, KPR04]. The variant of approximated distance labeling schemes has been considered in [GKK⁺01, Tho01, Tho04, TZ01, TZ05, Sli05, GKR05, MHP05, Tal04]. Efficient distance labeling schemes are known for a number of restricted graph families. $\mathcal{O}(\sqrt{n} \log n)$ bit labels are enough for planar graph, which is not tight with respect to the $\Omega(n^{1/3})$ lower bound. Similarly for bounded degree graph, $\Omega(\sqrt{n})$ is a lower bound on the label length. Finally $\mathcal{O}(\log^2 n)$ -distance labeling schemes are known for interval and permutation graphs [KKP00], for distance hereditary graphs [GP03a], for bounded tree-width graphs and more generally for bounded clique-width graphs [CV03].

A trivial observation is that a distance labeling scheme for a family \mathcal{F} requires labels at least as large as labels of an adjacency labeling scheme for \mathcal{F} . A first natural question is: 1) *could it be the same size for a large enough graph family?* As shown in [KNR92], information-theoretic lower bound coming from the number of n -vertex graphs in the family plays an important role for the label length. The unsolved *implicit graph representation conjecture* of [KNR92] asks whether any hereditary family with $2^{\mathcal{O}(n \times f(n))}$ labeled graphs of n vertices enjoys a $\mathcal{O}(f(n))$ -adjacency labeling ($\Omega(f(n))$ is a lower bound). Moreover, up to now, none of the hereditary graph families is known to support an $o(\log^2 n)$ -distance labeling scheme². It should be noticed that among the graphs families listed above, the $\Omega(\log^2 n)$ lower bound on label size for n -vertex trees does not apply neither to interval nor to permutation graphs. So the second question is: 2) *can we improve the $\mathcal{O}(\log^2 n)$ upper bound [KKP00] for either interval or permutation graphs?* This paper positively answers to the above questions by considering the family of interval graphs. Using some idea of this paper, [BG05b, BG05a] positively answer to the second case.

A graph $G = (V, E)$ is an *interval graph* (see e.g. [BLS99]) iff it is the intersection graph of a family \mathcal{I} of intervals on the real line: vertices of V are in bijection with the intervals

¹In this paper all the logarithms are in base two.

²It 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.

of \mathcal{I} and two vertices x and y are adjacent iff their intervals $\mathcal{I}(x)$ and $\mathcal{I}(y)$ intersect. \mathcal{I} is a *realizer* of the interval graph G . Obviously storing the bounds of each intervals defines an $\mathcal{O}(\log n)$ adjacency labeling scheme, as it can easily be shown that all the boundaries of \mathcal{I} can be chosen in $\{1 \dots 2n\}$. A graph is a *proper interval graph* iff it is the intersection graph of an inclusion-free family of intervals, i.e. none of the intervals is contained in another interval.

In this paper, we show that any $\ell(n)$ -distance labeling on the n -vertex interval graph family requires some labels of length $\ell(n) \geq 3 \log n - \mathcal{O}(\log \log n)$ bits. This lower bound derives from an new estimator of the number of interval graphs. We also obtain a tight upper bound of $5 \log n + 3$ bits for the label length. To that aim, we reduce any interval graph G to a proper interval graph G' such that the distance in G between any two vertices x, y can be retrieved from the distance in G' and the adjacency in G between x and y . It surprisingly follows that the core of the distance labeling problem of interval graphs consists in the design of a $2 \log n$ -distance labeling scheme for proper interval graphs. We also prove that the bound for proper interval graph is tight. Moreover, once the labels have been assigned, the distance computation from the labels takes a constant number of additions and comparisons on $\mathcal{O}(\log n)$ bit integers. Even more interesting, if the list of intervals of the graph is given sorted, then the preprocessing step to set all the labels runs optimally in $\mathcal{O}(n)$ time. Our scheme extends to circular-arc graphs.

At this step, it is worth to remark that any $\ell(n)$ -distance labeling scheme on a family \mathcal{F} converts trivially into a non-distributed data-structure for \mathcal{F} of $\mathcal{O}(\ell(n)n/\log n)$ space. The time complexity of distance queries remains unchanged, being assumed that a cell of space can store $\Omega(\log n)$ bits of data. Therefore, as a byproduct we can compute in $\mathcal{O}(n)$ time an $\mathcal{O}(n)$ space data-structure for interval graphs supporting constant time distance queries. This latter formulation implies the result of [CLSS98]. However, both approaches differ in essence. The technique of [CLSS98] 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). If $l(x) > l(y) + 1$ then $\text{dist}(x, y) = l(x) - l(y) - 1 + d_1(z, x)$ where $z = A(l(x) - l(y) - 1, x)$, and where $d_1(z, x)$ is the distance between two nodes whose levels differ by at most 1. The distance d_1 is 1, 2 or 3 depending on the interval representation of the involved vertices. Answering query is mainly based on the efficient parallel implementation of *level ancestor queries* on trees (to compute z) of [BV94]. However, this clever scheme cannot be converted into a distributed data-structure as ours 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 label. 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 approach of [CLSS98].

Section 2 deals with lower bounds. We first show that the number $I(n)$ of n -vertex labeled interval graphs is at least $2^{2n \log n - o(n \log n)}$ which is tight. This first bound is then

used to prove the lower bound on the length label for any distance labeling scheme of n -vertex interval graphs. A lower bound for proper interval graphs is also given. Section 3 describes a $(5 \log n + 3)$ -distance labeling scheme for interval graphs and its extension to circular arc graphs. Section 4 deals with the relationship between distance labeling schemes and the notion of universal distance matrix.

2 Lower bounds

This section establishes lower bounds on the label length required to get an exact distance labeling scheme for interval graphs and proper interval graphs. We first prove a lower bound on the number of interval graphs from which, using the technical composition tool of α -linkable subfamily (introduced in Section 2.2), we derive the announced lower bounds on distance labeling schemes.

2.1 On the number of interval graphs

Let $I(n)$ be the number of interval graphs with n vertices. Computing $I(n)$ is an unsolved graph-enumeration problem. Cohen, Komlós and Muller [CKM79] gave 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

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

As the total number of labeled n -vertex m -edge graphs is $\binom{n}{m}$, it follows a formula of $p(n, m) \cdot \binom{n}{m}$ for the number of labeled interval graphs with $m = \Theta(n^{5/6})$ edges. Unfortunately, using this formula it turns out that $I(n) \geq 2^{\Omega(n^{5/6} \log n)} = 2^{o(n)}$, 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 2.1 *The number $I(n)$ of labeled n -vertex connected interval graphs satisfies $\frac{1}{n} \log I(n) \geq 2 \log n - \log \log n - \mathcal{O}(1)$. It follows that there are $2^{\Omega(n \log n)}$ unlabeled n -vertex interval graphs.*

Proof. There are at least $I(n)/n!$ unlabeled interval graphs. First let us show that the lower bound on $\frac{1}{n} \log I(n)$ stated in Theorem 2.1 implies that there are $2^{\Omega(n \log n)}$ unlabeled n -vertex interval graphs. Indeed, using the fact that $\log(n!) = n \log n - \Theta(n)$, we have:

$$\begin{aligned}
\frac{1}{n} \log I(n) &\geq 2 \log n - \log \log n - \mathcal{O}(1) \\
\Rightarrow \log I(n) &\geq 2n \log n - n \log \log n - \mathcal{O}(n) \\
\Rightarrow \log I(n) - \log(n!) &\geq 2n \log n - \log(n!) - n \log \log n - \mathcal{O}(n) \\
\Rightarrow \log(I(n)/n!) &\geq n \log n - n \log \log n - \mathcal{O}(n) \\
\Rightarrow I(n)/n! &\geq 2^{n \log n - o(n \log n)}.
\end{aligned}$$

So there are $2^{\Omega(n \log n)}$ unlabeled interval graphs. Let us now prove the lower bound on $I(n)$.

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

From every $S = (S_1, \dots, S_p) \in \mathcal{S}_{p,k}$ we construct an interval graph G_S of vertex set $V(G_S) = A \cup B \cup C$ with: $A = \{a_{i,j} \mid 1 \leq i \leq p, 1 \leq j \leq k\}$, $B = \{b_0, \dots, b_p\}$, and $C = \{c_1, \dots, c_k\}$. 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 $\mathcal{I}(v)$ and defined as follows: for $i \in \{1, \dots, p\}$ and $j \in \{1, \dots, k\}$ (in the following, we denote by $s_{i,j}$ the j -th smallest 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. 1 (where $p = k = 3$), for each fixed i , the right boundaries of the $a_{i,j}$'s must differ, and must fall into the area delimited by the two vertical dotted lines. The intervals associated with the b_i 's and c_j 's are all of length k .

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 $j \in \{1, \dots, k\}$. Clearly, $\mathcal{F}_{p,k}$ is composed of labeled connected interval graphs on $n = (p+1)(k+1)$ vertices. Thus, $I((p+1)(k+1)) \geq |\mathcal{F}_{p,k}|$. We will first show that:

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

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, \dots, c_k are labeled $1, \dots, k+1$. Hence $|\mathcal{F}_{p,k}| \leq (p(k+1))! |\mathcal{S}_{p,k}|$. To prove equality, we describe hereafter a two-step procedure that from any $G \in \mathcal{F}_{p,k}$: 1) retrieves in a unique way a sequence S

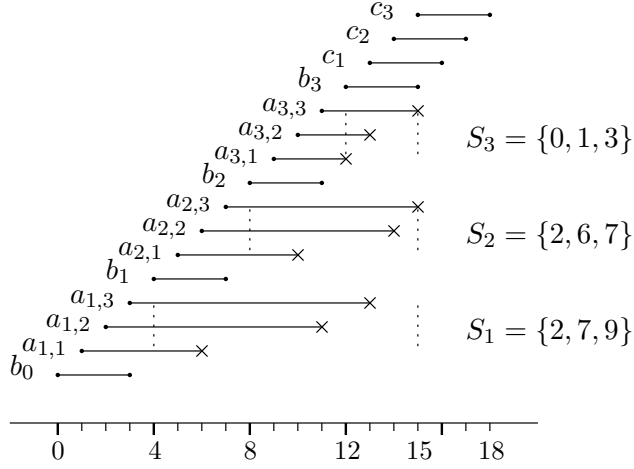


Figure 1: The interval representation of G_S for $S = (S_1, S_2, S_3)$.

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 $|\mathcal{F}_{p,k}| \geq (|A| + |B \setminus \{b_0\}|)! |\mathcal{S}_{p,k}|$.

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* if one can decide whether $v = a_{i,j}$ or $v = b_i$ or $v = c_j$ (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 eventually its associated integer $s_{i,j}$, completing Step 1 as well.

For every $i \in \{0, \dots, p\}$, let L_i denote the set of vertices of G_S whose left boundary is $\leq i(k+1)$, 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 $L_i \setminus L_{i-1} = \{a_{i,1}, \dots, a_{i,k}\} \cup \{b_i\}$.

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 $a_{i,1}, \dots, a_{i,k}$;
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} are 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 $b_{i-1}, a_{i,1}, \dots, a_{i,k}$, it follows (combining Property 1 and 2 to G_{i+1}) that b_i can be identified: it is the only degree k vertex of $G_{i+1} \setminus C$. Thus L_i 's vertices 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 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 $\binom{a}{b} \geq (a/b)^b$ we have:

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

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

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

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

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

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

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

This ends the proof of Theorem 2.1. □

It is clear that $\frac{1}{n} \log I(n) \leq \lceil 2 \log(2n) \rceil = 2 \log n + \mathcal{O}(1)$, as each vertex can be described by its interval representation (there are at most $2n$ distinct interval boundaries). So the lower bound of Theorem 2.1 on $\frac{1}{n} \log I(n)$ is tight up to an additive factor of $\log \log n$.

2.2 Lower bounds on distance labeling schemes

The eccentricity of a vertex x in a graph G , denoted $\text{ecc}_G(x)$, is the maximum of the distances between x and any vertex of G . An α -graph, for integer $\alpha \geq 1$, is a graph G

having a pair of vertices (l, r) , possibly with $l = r$, such that $\text{ecc}_G(l) \leq \alpha$ and $\text{ecc}_G(r) \leq \alpha$. Let $\mathcal{S} = (H_0, H_1, \dots, H_k)$ be a sequence of α -graphs, and let (l_i, r_i) denote the pair of vertices that defines the α -graph H_i , for $i \in \{0, \dots, k\}$. For each non-null integer sequence $W = (w_1, \dots, w_k)$, we denote by \mathcal{S}^W the graph obtained by attaching a path of length w_i between the vertices r_{i-1} and l_i , for every $i \in \{1, \dots, k\}$ (see Fig. 2).

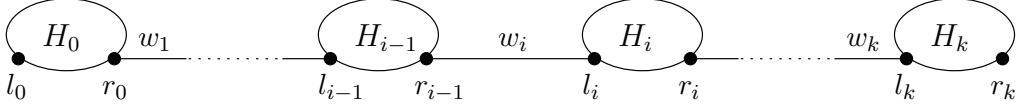


Figure 2: Linking a sequence of α -graphs.

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

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

Lemma 2.1 *Let \mathcal{H} be an α -linkable sub-family of a graph family \mathcal{F} . Let $H(N)$ denote the number of labeled N -vertex graphs of \mathcal{H} . Then, every distance labeling scheme on \mathcal{F}_n requires a label of length at least $\frac{1}{N} \log H(N) + \log N - 9$, where $N = \lfloor n/(\alpha \log n) \rfloor$.³*

Let us first sketch the proof. It uses a sequence \mathcal{S} of $k = \Theta(\log n)$ graphs H_i taken from an arbitrary α -linkable sub-family \mathcal{H} . Each graph H_i has $N = \Theta(n/\log n)$ vertices and consecutive H_i 's from \mathcal{S} are spaced with paths of length $\Theta(n/\log n)$. Intuitively, the term $\frac{1}{N} \log H(N)$ measures the minimum label length required to compute the distance between any two vertices within a same H_i 's (i.e., whether they are adjacent or at distance at least two). The $\log N$ term is required to compute the distance between vertices of distinct H_i 's. The difficulty is to show that some vertices require both informations, observing that one can distribute information on the vertices in a non trivial way. For instance, the two extreme 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 labels for a distance query.

Proof. (of Lemma 2.1) Let p, q, k be non-null integers. Let \mathcal{W} be the set of all the integer sequences (w_1, \dots, w_k) such that $w_i \in \{1 + (j-1)(2\alpha+1) \mid 1 \leq j \leq q\}$. Since a sequence may contain repeated elements of \mathcal{W} , $|\mathcal{W}| = q^k$. We consider \mathcal{H} be an arbitrary α -linkable sub-family of \mathcal{F} . Let $H(p)$ be the number of labeled p -vertex graphs of \mathcal{H} . For every $W \in \mathcal{W}$, we denote by $\mathcal{H}_p(W)$ the set of all graphs \mathcal{S}^W where \mathcal{S} is a sequence of $k+1$ graphs of \mathcal{H} , each with exactly p vertices (see Fig. 2).

³ \mathcal{F}_n denotes the family of graphs belonging to \mathcal{F} with at most n vertices.

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

Let $\langle L, f \rangle$ be any distance labeling scheme on \mathcal{F}_n . We consider the set \mathcal{L}_W of all the $(k+1)$ -tuples of labels for all the graphs $\mathcal{S}^W \in \mathcal{H}_p(W)$, each tuple being formed by taken a vertex label successively in H_0, H_1, \dots, H_k . More formally,

$$\mathcal{L}_W = \bigcup_{\substack{\mathcal{S}^W \in \mathcal{H}_p(W) \\ \text{with } H=(H_0, \dots, H_k)}} \{(L(u_0, H_0), \dots, L(u_k, H_k)) \mid (u_0, \dots, u_k) \in V(H_0) \times \dots \times V(H_k)\}$$

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 the size of any label assigned by L on some graph of \mathcal{F}_n is at most t bits. Then, as there are $2^{t+1} - 1$ binary strings of length at most t , there are $(2^{t+1} - 1)^{k+1}$ distinct $(k+1)$ -tuples of such labels. Since \mathcal{L} is a set of $(k+1)$ -tuples, thus $|\mathcal{L}| \leq (2^{t+1} - 1)^{k+1}$, implying that $t \geq \log \left(|\mathcal{L}|^{1/(k+1)} + 1 \right) - 1 \geq \frac{1}{k+1} \log |\mathcal{L}| - 1$.

Claim 2.3 $|\mathcal{L}_W|^p \geq H(p)^{k+1}$.

We can assume each graph $H_j \in \mathcal{S}$ of any $\mathcal{S}^W \in \mathcal{H}_p(W)$ is defined on a copy of the same sorted set of vertices. Therefore a set $L_{\mathcal{S}}$ of $(k+1)$ -tuples can be associated with each \mathcal{S}^W

$$L_{\mathcal{S}} = \{(L(v_j, H_0), \dots, L(v_j, H_k)) \mid j \in [1, p]\},$$

each $(k+1)$ -tuple being composed by the labels of an arbitrary vertex in each H_i . We have $\bigcup_{\mathcal{S}^W \in \mathcal{H}_p(W)} L_{\mathcal{S}} \subseteq (\mathcal{L}_W)^p$. Let us show that $L_{\mathcal{S}} \neq L_{\mathcal{S}'}$ for all $\mathcal{S} \neq \mathcal{S}'$. Indeed, if $\mathcal{S} \neq \mathcal{S}'$, there exist two vertices, say v_{j_1} and v_{j_2} , adjacent in the i -th graph H_i of \mathcal{S} but not adjacent in $H'_i \in \mathcal{S}'$ ($\text{dist}_{\mathcal{S}'^W}(v_{j_1}, v_{j_2}) \geq 2$). The distance decoder f must therefore return different values for the corresponding pair of labels, i.e., $f(L(v_{j_1}, H_i), L(v_{j_2}, H_i)) \neq f(L(v_{j_1}, H'_i), L(v_{j_2}, H'_i))$. As both labels, $L(v_{j_1}, H_i)$ and $L(v_{j_1}, H'_i)$ ($L(v_{j_2}, H_i)$ and $L(v_{j_2}, H'_i)$ respectively), are located in the same respective $(k+1)$ -tuple of $L_{\mathcal{S}}$ and $L_{\mathcal{S}'}$, $L_{\mathcal{S}} \neq L_{\mathcal{S}'}$ follows. It implies that

$$\left| \bigcup_{\mathcal{S}^W \in \mathcal{H}_p(W)} L_{\mathcal{S}} \right| \geq |\mathcal{H}_p(W)|.$$

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

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

First observe that if W and W' differ at the i -th entry, with $w_i \neq w'_i$, then $[w_i, w_i + 2\alpha] \cap [w'_i, w'_i + 2\alpha] = \emptyset$. Indeed, by construction of the integers of \mathcal{W} , $|w_i - w'_i| \geq 2\alpha + 1$. We observe that $f(L(u, H_{i-1}), L(v, H_i)) \in [w_i, w_i + 2\alpha]$, for every $i \in \{1, \dots, k\}$ and for all $(u, v) \in V(H_{i-1}) \times V(H_i)$. Note also that every tuple of labels $T \in \mathcal{L}_W$ contains a label taken from H_{i-1} and a label from H_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 and since $|\mathcal{W}| = q^k$, $|\mathcal{L}| \geq \sum_{W \in \mathcal{W}} |\mathcal{L}_W| \geq H(p)^{(k+1)/p} q^k$ holds. From Claim 2.2, the label length is at least (note that $k/(k+1) \geq 1 - 2/(k+2)$ for $k \geq 1$)

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

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

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

□

Let us now show that interval graphs and proper interval graphs both enjoys a large enough α -linkable subfamily of labeled graphs. The lower bounds on distance labeling schemes for these two graph families therefore follows from Lemma 2.1 and Theorem 2.1.

Theorem 2.5 *Any distance labeling scheme on the family of n -vertex interval graphs requires a label of length at least $3 \log n - 4 \log \log n$.*

Proof. Let $I(N)$ be the number of labeled interval graphs with N vertices. Let \mathcal{H} be that family of labeled N -vertex interval graphs enjoying a universal vertex where the universal vertex is labeled N . Any graph of \mathcal{H} is a 1-graph (set the universal vertex as $l = r$). And by the way \mathcal{H} is a 1-linkable subfamily of interval graphs. Moreover it is straightforward to see that any $(N-1)$ -vertex labeled interval graph is an induced subgraph of some $H \in \mathcal{H}$. It follows that the number $H(N)$ of N -vertex labeled graphs of \mathcal{H} is precisely $I(N-1)$.

By Lemma 2.1 applied to \mathcal{H} and the interval graphs family the length t of the maximum label is at least:

$$t \geq \frac{1}{N} \log H(N) + \log N - \mathcal{O}(1) \geq \frac{1}{N} \log I(N-1) + \log N - \mathcal{O}(1)$$

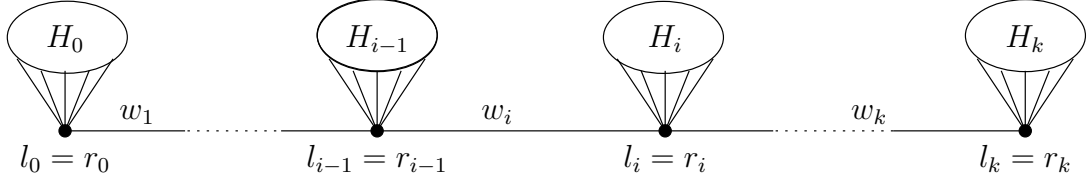


Figure 3: Each H_i is a 1-linkable N -interval graph with a universal vertex $l_i = r_i$

From Theorem 2.1, we have:

$$\begin{aligned} \frac{1}{N} \log I(N-1) &= \frac{N-1}{N} (2 \log(N-1) - \log \log(N-1) - \mathcal{O}(1)) \\ &\geq 2 \log(N-1) - \log \log N - \mathcal{O}(1) \end{aligned}$$

Since \mathcal{H} is a 1-graph family, we have $\alpha = 1$. Therefore Lemma 2.1 sets $N = \lfloor n / \log n \rfloor$. It implies that $N-1 \geq \frac{n}{2 \log n}$.

$$\begin{aligned} \frac{1}{N} \log I(N-1) &\geq 2 \log\left(\frac{n}{2 \log n}\right) - \log \log \lfloor n / \log n \rfloor - \mathcal{O}(1) \\ &\geq 2 \log n - 3 \log \log n + \log \log \log n - \mathcal{O}(1) \\ &\geq 2 \log n - 3 \log \log n \end{aligned}$$

Gathering everything together, we obtain:

$$\begin{aligned} t &\geq 2 \log n - 3 \log \log n + \log \lfloor n / \log n \rfloor \\ &\geq 3 \log n - 4 \log \log n \end{aligned}$$

Completing the proof. □

Theorem 2.6 *Any distance labeling scheme on the family of n -vertex proper interval graphs requires a label of length at least $2 \log n - 2 \log \log n - \mathcal{O}(1)$.*

Proof. Let H_{2p} be the intersection graph of the interval family $\{\mathcal{I}(x_i) = [i, i+p], \mathcal{I}(y_i) = [i+p, i+2p-1], 1 \leq i \leq p\}$. Since any interval is of length p , H_{2p} is a proper interval graph.

Let \mathcal{H} be the family of labeled graphs H isomorphic to H_{2p} 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 have been identified in H , then x_{i+1} and y_{i+1} can be identified too. It suffices to remove all the vertices x_1, \dots, x_{i-1} and y_1, \dots, y_{i-1} from H (and so to form a graph isomorphic to $H_{2p-2(i-1)}$), 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, the cardinality of \mathcal{H} is, for $N = 2p$, $H(N) = (N-2)!$.

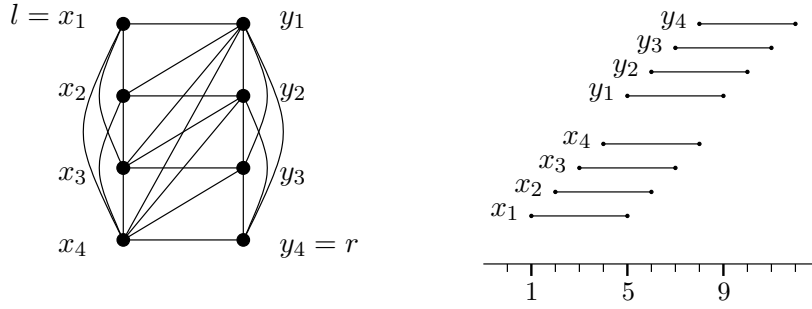


Figure 4: The graph H_4 and its interval representation.

Now, setting $l = x_1$ and $r = y_p$, l and r are both of eccentricity 2, as the vertex x_p is a universal vertex. So \mathcal{H} is a family of 2-graphs. Moreover \mathcal{H} is a 2-linkable subfamily of the proper interval graphs. Since $H(N) = (N - 2)!$, with $N = 2p$. We note that $\frac{1}{N} \log(N - 2)! \geq \log N - \mathcal{O}(1)$. By Theorem 2.1 and Lemma 2.1, for $N = \lfloor n/(2 \log n) \rfloor$, the length of the maximum label is at least:

$$\frac{1}{N} \log H(N) + \log N - \mathcal{O}(1) \geq \log N + \log N - \mathcal{O}(1) \geq 2 \log n - 2 \log \log n - \mathcal{O}(1)$$

ending the proof. \square

3 Upper bounds on distance labeling schemes

This section deals with optimal distance labeling schemes for proper interval graphs, interval graphs and circular-arc graphs. We first show how to complete any interval graph G into a proper interval graph G' so that the distance between two vertices x and y in G can be retrieved from the distance in G' and the adjacency in G between x and y . Then, we present a distance labeling scheme for proper interval graphs. This scheme is based on an adjacency labeling scheme of an auxiliary graph. It turns out that the auxiliary graph of any proper interval graph enjoys an optimal adjacency labeling scheme. The $\mathcal{O}(\log n)$ distance labeling scheme for interval graphs follows. Finally, an extension of this scheme is proposed for the family of circular-arc graphs.

From now on, any interval graph is supposed to be an n -vertex connected graph. Let G be an interval graph, then $L(x)$ and $R(x)$ respectively denote 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 boundaries. We will also assume without loss of generality, that the left boundaries are pairwise distinct. This hypothesis is not restrictive since in $\mathcal{O}(n)$ time one can scan all the boundaries from $\min_x L(x)$ to $\max_x L(x)$ and compute another family with sorted and distinct boundaries⁴.

⁴Another technical assumption is that all the boundaries of the realizer are positive integers bounded by

3.1 Reduction to proper interval graphs

There are several well known characterizations of proper interval graphs: this is exactly the family of $K_{1,3}$ -free interval graphs [Weg67]; G is a proper interval graph iff it has an interval representation using a family of unit length intervals [Rob69]. Proper interval graph recognition algorithms run in linear time ($\mathcal{O}(n+m)$ where n is the number of vertices and m the number of edges) [CKN⁺95, dFMdM95, HBJH90]. Moreover these algorithms output an inclusion-free realizer.



Figure 5: The $K_{1,3}$ is an interval graph but not a proper interval graph.

Definition 3.1 Let \mathcal{R} be the realizer of an interval graph. If any pair of intervals \mathcal{I} and \mathcal{I}' , such that $\mathcal{I} \subseteq \mathcal{I}'$, share their right boundary (ie. $R = R'$), then \mathcal{R} is a proper realizer.

Remark: Let us notice that an inclusion-free realizer of intervals is a proper realizer. Moreover if \mathcal{R} be a proper realizer and $\{a, b, c\}$ be a stable set such that $L(a) < R(a) < L(b) < R(b) < L(c) < R(c)$. Then a and c do not have any common neighbor. Otherwise the interval $\mathcal{I}(x)$ of such a common neighbor x would satisfy $L(b) \subset L(x)$ and $L(b) \neq L(x)$: contradiction. It follows that a graph is a proper interval graph iff it enjoys a proper realizer.

Definition 3.2 Let \mathcal{R} be a realizer of an interval graph $G = (V, E)$. Let $x \in V$ such that $\mathcal{I}(x) \subset \mathcal{I}(y)$ and $R(x) < R(y)$ for some vertex y , then the enclosing neighbor of x , denoted by $N_e(x)$, is the neighbor of x such that:

- $\mathcal{I}(x) \subset \mathcal{I}(N_e(x))$;
- $\mathcal{I}(N_e(x))$ is maximal for the inclusion and
- $R(N_e(x))$ is maximum.

As the left boundaries of \mathcal{I} are pairwise disjoint, the enclosing neighbor, if it exists, is unique for every vertex. A proper realizer \mathcal{R}' can be obtained from \mathcal{R} as follows: if x has an enclosing neighbor, then we set $\mathcal{I}'(x) = [L'(x) = L(x), R'(x) = R(N_e(x))]$, and we

some polynomial of n , so that boundaries can be manipulated in constant time on RAM-word computers. Note that linear time recognition algorithms produce such layout.

set $\mathcal{I}'(x) = \mathcal{I}(x)$ otherwise. Let us remark: 1) $R'(x) \geq R(x)$; 2) by maximality of its right boundary $\mathcal{I}(N_e(x)) = \mathcal{I}'(N_e(x))$; and 3) G is a subgraph of G' and thus $E \subseteq E'$. We say that a vertex x has been *extended* in G' if $R'(x) > R(x)$.

Lemma 3.1 *The realizer \mathcal{R}' is a proper realizer. It can be computed from \mathcal{R} in $\mathcal{O}(n)$ time.*

Proof. Assume there is a pair of vertices u and v such that $\mathcal{I}'(u) \subset \mathcal{I}'(v)$ and $R'(u) < R'(v)$. W.l.o.g. assume $\mathcal{I}'(v)$ is maximal for the inclusion. Notice that $R(v) = R'(v)$, otherwise v would have an enclosing neighbor w that would imply $\mathcal{I}'(v) \subset \mathcal{I}'(w)$: contradicting the maximality of v . $R(u) \leq R'(u)$, we have $L(v) < L(u) < R(u) < R'(v)$. $R(v) = R'(v)$ implies $\mathcal{I}(u) \subset \mathcal{I}(v)$: i.e., u has an enclosing neighbor and therefore $R'(u) \geq R(v) = R'(v)$: a contradiction with $R'(u) < R'(v)$.

The realizer \mathcal{R}' can be computed as follows: scan the intervals of \mathcal{R} accordingly to their left boundary ordering and output the ordered set of vertices $S = \{x_1, \dots, x_i, \dots\}$ so that x_{i+1} has minimum left boundary among $\{x \mid R(x_i) < R(x) \text{ and } L(x_i) < L(x)\}$. Now for any vertex $v \notin S$, its enclosed neighbor is the maximum $x_i \in S$ such that $L(x_i) < L(v)$. Since two scans of the vertices suffice to compute those enclosed neighbors, computing \mathcal{R}' only requires $\mathcal{O}(n)$ time. \square

The main result of the subsection shows that if the realizer \mathcal{R} of G is not a proper realizer, then the distances in G can be retrieved from the distances in G' whose realizer \mathcal{R}' is proper. Thereby we prove that distance labeling problem of interval graph reduces to proper distance labeling problem of proper interval graph. To that end, an extra definition is needed.

Definition 3.3 *For any vertex x , the maximum neighbor of x , denoted by $N_m(x)$, is the neighbor of x such that $R(N_m(x))$ is maximum.⁵*

Like enclosing neighbors, maximum neighbors are not extended in G' . Otherwise we would have $\mathcal{I}(N_m(x)) \subset \mathcal{I}(v)$ for some vertex v and $R(N_m(x)) < R(v)$. Since $\mathcal{I}(x) \cap \mathcal{I}(N_m(x)) \neq \emptyset$, we also have $\mathcal{I}(x) \cap \mathcal{I}(v) \neq \emptyset$. Since $R(N_m(x)) < R(v)$, by definition, we should have $v = N_m(x)$: contradiction. To prove Theorem 3.4, the above lemma is required.

Lemma 3.2 *If x and y are two vertices such that $R(x) = R'(x) \leq R(y)$. Then $\text{dist}_G(x, y) = \text{dist}_{G'}(x, y)$.*

⁵The set of maximum neighbors of each vertex can be computed in $\mathcal{O}(n)$ time. Since \mathcal{R}' is proper, the maximum neighbor of a given vertex x is precisely the vertex y with maximum left boundary satisfying $L(y) \leq R(x)$. Since we made the assumption that the intervals have been given sorted w.r.t. their left boundaries, a single scan enables to compute the maximum neighbor of each vertex.

Proof. Let us first remark that since G is a subgraph of G' , then $\text{dist}_G(x, y) \geq \text{dist}_{G'}(x, y)$. So assume that $\text{dist}_G(x, y) = k > 1$. Let us first remark that any path of G' that does not use any extended vertex is also a path of G . Let $P = [x = u_0, u_1, \dots, u_j, \dots, u_k = y]$ be a shortest x, y -path in G' using a minimum number of extended vertices. Assume there exists some $j > 0$ such that u_j has been extended (i.e., $\mathcal{I}(u_j) \neq \mathcal{I}'(u_j)$) and choose j minimum. By assumption $\mathcal{I}(u_{j-1}) = \mathcal{I}'(u_{j-1})$ holds and implies that u_{j-1} and u_j are adjacent in G . Since $\mathcal{I}(u_j) \subset \mathcal{I}(N_e(u_j))$, substituting in P vertex u_j by vertex $N_e(u_j)$ enlightens a shortest x, y -path P' in G' using fewer extended vertices than P : contradiction, P does not contain any extended vertex. \square

Theorem 3.4 *Let x and y be two vertices such that $R(x) \leq R(y)$. Then,*

$$\text{dist}_G(x, y) = \text{dist}_{G'}(N_m(x), y) + 1 - \text{adj}_G(x, y) . \quad (2)$$

Proof. If x and y are adjacent, then $N_m(x)$ and y are also adjacent in both G and G' , therefore the result holds. So assume that x and y are not adjacent. Since $N_m(x)$ has not been extended in G' , by Lemma 3.2, $\text{dist}_{G'}(N_m(x), y) = \text{dist}_G(N_m(x), y)$. So we have to prove that $\text{dist}_G(x, y) = 1 + \text{dist}_G(N_m(x), y)$.

Let us first remark that $\text{dist}_G(N_m(x), y) \leq \text{dist}_G(x, y)$ since $R(x) < L(y)$ and $R(x) < R(N_m(x))$. The only case to be considered is $\text{dist}_G(N_m(x), y) = k > 1$. Let $P = [N_m(x), v_1, \dots, v_{k-1}, y]$ be a shortest $N_m(x), y$ -path. Let us assume that $\text{dist}_G(x, y) = k$, then there exists a path $Q = [x, u_1, u_2, \dots, u_{k-1}, y]$ of length k . By definition of $N_m(x)$, we have $R(u_1) < R(N_m(x))$. It implies that $N_m(x)$ is adjacent to u_2 , therefore $[N_m(x), u_2, \dots, u_{k-1}, y]$ is a path of length $k - 1$: contradiction. \square

Given a distance labeling scheme of a proper interval graph, Eq. (2) directly describes the decoder function of a distance labeling scheme for the family of interval graphs. The label $L(x, G)$ of each vertex should contain:

- $R(x)$ and $L(x)$, the boundaries of $\mathcal{I}(x)$ that enable us to test $\text{adj}_G(x, y)$;
- $L'(x, G')$ and $L'(N_m(x), G')$, the labels of x and $N_m(x)$ in the distance labeling scheme of the proper interval graph G' that enable us to compute $\text{dist}_{G'}(x, y)$.

Corollary 3.5 *If there exists a distance labeling scheme for n -vertex proper interval graphs using labels of size $s(n)$, then n -vertex interval graphs family enjoys a distance labeling scheme using labels of size $2\lceil \log 2n \rceil + 2s(n)$.*

3.2 Distance labeling scheme of proper interval graphs

Let us now assume that a proper realizer (of a proper interval graph) is given. Let x_0 be the vertex with minimum left boundary. Notice x_0 also has a minimum right boundary.

We define the *layer partition* V_0, V_1, \dots, V_k as the partition of vertices into distance layers from x_0 :

$$\forall i \geq 0, V_i = \{v \mid \text{dist}(x_0, v) = i\}$$

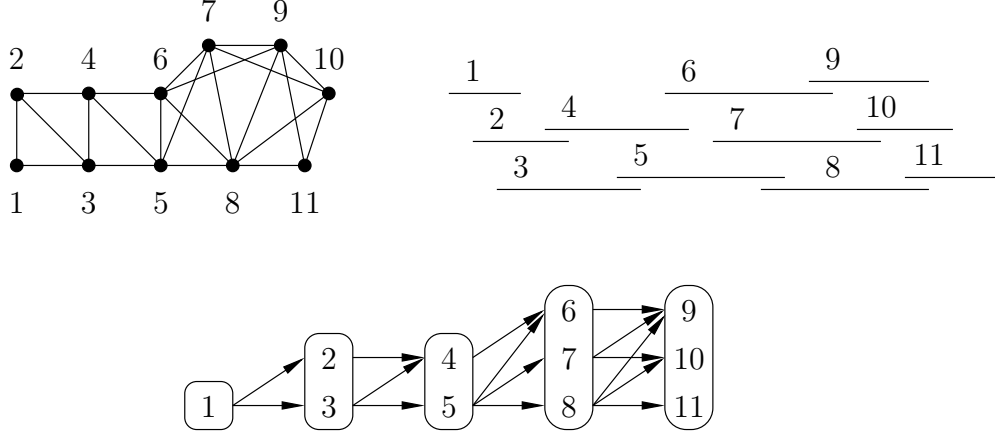


Figure 6: A proper interval graph with an interval representation and the associated layer partition.

Given the sorted list of intervals, 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 acyclic.

Claim 3.6 *The transitive closure H^t of H is a poset (partially ordered set).*

Let $\text{adj}_{H^t}(x, y)$ be the boolean function such that:

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

Notice that since H^t is a digraph, $\text{adj}_{H^t}(x, y) = 1$ implies that $\text{adj}_{H^t}(y, x) = 0$. Also by definition of H^t , if $\text{adj}_{H^t}(x, y) = 1$, then $\lambda(x) < \lambda(y)$ holds.

Theorem 3.7 *For all distinct vertices x and y such that $\lambda(x) \leq \lambda(y)$,*

$$\text{dist}_G(x, y) = \lambda(y) - \lambda(x) + 1 - \text{adj}_{H^t}(x, y) \quad (3)$$

To prove Theorem 3.7, we need the following preliminary result.

Lemma 3.3 *$\forall i \geq 1$, let x_{i-1} be the vertex of V_{i-1} having maximum right boundary⁶. Then $\{x_{i-1}\} \cup V_i$ is a clique of G .*

⁶For $i > 0$, x_i is the maximum neighbor of x_{i-1} and $x_0 \in V_0$

Proof. Let us first show that x_{i-1} dominates V_i . It holds for $i = 1$ by definition of the layer partition. Let us consider the case for larger i . First any vertex $v \in V_i$ has a neighbor in V_{i-1} and x_{i-1} has a maximum right boundary among V_{i-1} 's vertices. It implies that $L(x_{i-1}) < L(v) < R(x_{i-1})$: in other words, v neighbors x_{i-1} .

Let us show now that V_i induces a clique. Since $V_0 = \{x_0\}$, for any vertex $v \in V_1$, $\mathcal{I}(v)$ contains $R(x_0)$, both V_0 and V_1 are cliques. For larger i , we already proved that any $v \in V_i$ satisfies $L(x_{i-1}) < L(v) < R(x_{i-1})$. Since the realizer of G is assumed to be proper, we also have $L(v) < R(x_{i-1}) \leq R(v)$. Therefore, for any $v \in V_i$, $\mathcal{I}(v)$ contains $R(x_{i-1})$. It implies that V_i is a clique. \square

Proof of Theorem 3.7. Let x and y be two vertices such that $\lambda(x) = i$ and $\lambda(y) = i + h$ with $h \geq 0$. It is obvious that $h \leq \text{dist}_G(x, y)$. Note that by definition of the layer partition, the statement holds for $i = 0$.

Let us consider $i > 0$. Lemma 3.3 implies the result in cases $h = 0$ and $h = 1$. So assume the statement holds for $h - 1 \geq 0$ and let us prove it for h . By Lemma 3.3, x is adjacent to x_i and x_{i+h-1} is adjacent to y . Therefore $[x, x_i, \dots, x_{i+h-1}, y]$ is a path of length $h + 1$ and $\text{dist}_G(x, y) \leq \lambda(y) - \lambda(x) + 1$. Inequality holds iff x is adjacent to some vertex $z \in V_{i+1}$ such that $\text{dist}_G(z, y) = h - 1$. In that case, we have $\text{adj}_{H^t}(x, z) = 1$ and by induction hypothesis $\text{adj}_{H^t}(z, y) = 1$ holds. Since H^t is transitive, $\text{dist}_G(x, y) = h$ iff $\text{adj}_{H^t}(x, y) = 1$. That ends the proof. \square

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 with the layer partition. Let us now prove that in H^t the adjacency can be encoded using short labels (namely $2 \log n$ bits labels).

Theorem 3.8 *There exists a total ordering π of the vertices such that*

$$\text{adj}_{H^t}(x, y) = 1 \text{ iff } \lambda(x) < \lambda(y) \text{ and } \pi(y) < \pi(x) \quad (4)$$

Moreover given H^t and \mathcal{F} , the ordering π can be computed in $\mathcal{O}(n)$ time.

Proof. We first claim that such an ordering π is computed by the following algorithm:

π is the pop ordering of a DFS on H using $L(x)$ 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. 6, we obtain the following linear ordering:

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

It is easy to check that if $\text{adj}_{H^t}(x, y) = 1$, then $\lambda(x) < \lambda(y)$ and $\pi(y) < \pi(x)$. So assume that $\text{adj}_{H^t}(x, y) = 0$ and $\lambda(x) < \lambda(y)$ (which implies $L(x) < L(y)$). Let us first show by contradiction that, for any z such that $\lambda(y) < \lambda(z)$, if $\text{adj}_{H^t}(x, z) = 1$, then $\text{adj}_{H^t}(y, z) = 1$. $\text{adj}_{H^t}(x, z) = 1$ implies the existence of an x, z -shortest path of length $\lambda(z) - \lambda(x)$. Let $[x, \dots, v_{\lambda(y)-1}, v_{\lambda(y)}, v_{\lambda(y)+1}, \dots, z]$ be such a path with $v_i \in V_i$, for $i \in \{\lambda(y) - 1, \lambda(y), \lambda(y) + 1\}$, and possibly $x = v_{\lambda(y)-1}$ and $z = v_{\lambda(y)+1}$. Notice that $\text{adj}_{H^t}(x, y) = 0$ implies that $\text{adj}_{H^t}(v_{\lambda(y)-1}, y) = 0$. Similarly $\text{adj}_{H^t}(y, z) = 0$ would imply that $\text{adj}_{H^t}(y, v_{\lambda(y)+1}) = 0$. Therefore the set $\{y, v_{\lambda(y)-1}, v_{\lambda(y)}, v_{\lambda(y)+1}\}$ would induce a $K_{1,3}$ of G : contradicting the assumption that 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 $\text{adj}_{H^t}(x, z) = 1$. If x is also a sink, since $L(x) < L(y)$ the priority rule implies $\pi(x) < \pi(y)$. Assume x is not a sink. Therefore there exists z such that $\text{adj}_{H^t}(x, z) = 1$. We already proved that such a vertex z satisfies $\lambda(z) \leq \lambda(y)$. Moreover, we show that $L(z) < L(y)$. Otherwise z and y would both belong to $V_{\lambda(y)}$ and a shortest path of length $\lambda(y) - \lambda(x)$ between x and z would imply a shortest path of same length between x and y : contradicting $\text{adj}_{H^t}(x, y) = 0$. Finally both z and y are sinks, but $L(z) < L(y)$ implies that z is popped before y . Recursively applying this argument enables us to prove that x will be popped before y by the algorithm and so $\pi(y) < \pi(x)$.

Let us now analyse the time complexity of the computation of the π ordering. Notice that 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 boundary of their interval. Since the layers and also the neighborhood in H of any vertex appear consecutively in this array, each vertex can be associated with its first and last neighbor in the array. The priority rule implies that when a vertex v is popped by the algorithm, any vertex $u \in V_{\lambda(v)}$ such that $L(u) < L(v)$ has already been popped. Assuming that the layer of any vertex is stored and that for each layer, the index of the last popped vertex is maintained, the next vertex to be pushed can be found in $\mathcal{O}(1)$. Therefore π can be computed in $\mathcal{O}(n)$ time. \square

The *dimension* of a poset P is the minimum number d of linear orderings ρ_1, \dots, ρ_d such that $x <_P y$ iff $x <_{\rho_i} y$ for every i . Next corollary follows immediately from Theorem 3.8.

Corollary 3.9 *H^t is a poset of dimension two.*

We now have the material to define the distance labeling scheme for proper interval graphs. 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. (3) and (4). Clearly, $\lambda(x), \pi(x) \in \{1, \dots, n\}$, so the main result follows:

Theorem 3.10 *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.*

3.3 Optimized distance labeling scheme for interval graphs

According to Corollary 3.5 and Theorem 3.10, the family of interval graphs enjoys a $(6 \log n + \mathcal{O}(1))$ -distance labeling scheme with constant time decoder. We shall see that these labels can be compacted to $(5 \log n + \mathcal{O}(1))$ bits. Let us recall that for any vertex x of an interval graph G , we have to know $R(x)$ and $L(x)$, the boundaries of the interval $\mathcal{I}(x)$, plus the labels of x and $N_m(x)$ in the extended proper interval graph G' .

The improvement is based on the simple observation that $\lambda(x)$ and $\lambda(N_m(x))$ differs at most by one. Indeed there is no edge in G between pair of vertices that does not belongs to the same distance layer or consecutive distance layers. Therefore a single bit, say $b(x)$, suffices to recover $\lambda(N_m(x))$ from $\lambda(x)$. Therefore the label of a vertex x can be defined as follows:

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

As $1 \leq L(x) < R(x) \leq 2n$, so $2 \lceil \log n \rceil + 2$ bits suffice to store the first two integers. The next four 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 3.11 *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.*

3.4 A Scheme for Circular-Arc Graphs

Circular-arc graphs are a natural generalization of interval graphs, in which vertices are mapped to arcs of a circle rather than intervals on the real line. This section shows that the distance labeling scheme problem for circular arc graphs reduces to the interval graphs case. Consider a circular-arc graph G . We can associate to every vertex x of G a range $I(x) = [\theta_l(x), \theta_r(x)]$ of angles where $\theta_l(x)$ and $\theta_r(x)$ are taken clockwise in $[0, 2\pi)$. The range $I(x)$ can be seen as a “cyclic” interval in which $[\theta_l(x), \theta_r(x)]$ for $\theta_l(x) > \theta_r(x)$ stands for $[\theta_l(x), 2\pi) \cup [0, \theta_r(x)]$. The vertices x and y are adjacent if $I(x) \cap I(y) \neq \emptyset$. See an example on Fig. 7.

A neighbor y of x is a *right neighbor* (resp. *left neighbour*) iff $\theta_l(y) \in I(x)$ and $\theta_r(y) \notin I(x)$ (resp. $\theta_l(y) \notin I(x)$ and $\theta_r(y) \in I(x)$). Notice that some neighbor of x may neither be a right nor a left neighbor. A path x_0, x_1, \dots, x_k is a *right-path* from x_0 to x_k if, for every $i \in \{0, \dots, k-1\}$, 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, \dots, x_k .

Lemma 3.4 *Every shortest path between non-adjacent vertices x and y is either a right-path from x to y , or a right-path from y to x .*

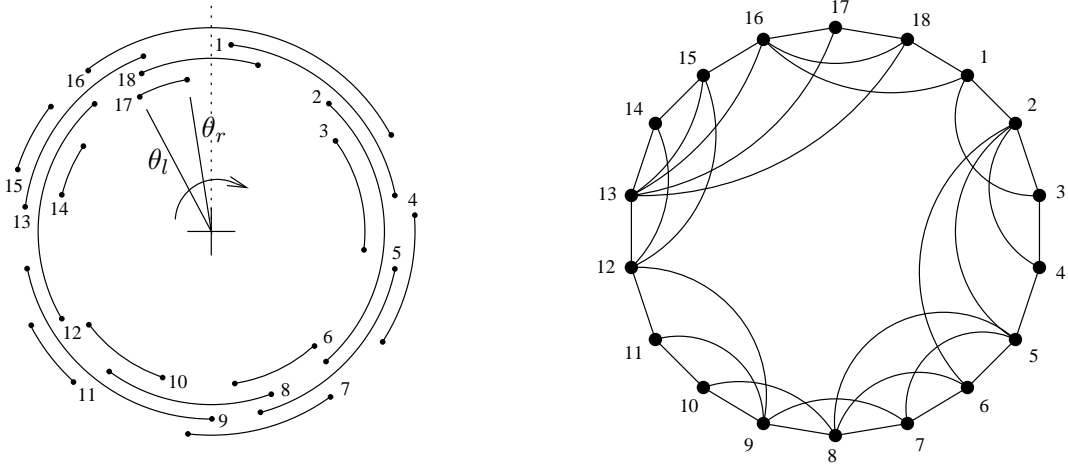


Figure 7: A circular-arc graph with its arc representation. Vertices are labeled from 1 to 18 according to their value θ_l .

Proof. Assume that $P = [x = x_0, x_1, \dots, x_k = y]$ 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 .

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 roles of θ_r and θ_l boundaries. Since the path from x_0 to x_k is not right path, there is an integer i , $1 \leq i < k$, 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.

- If x_{i+1} is not a left neighbor of x_i , then either $I(x_i)$ is included in $I(x_{i+1})$ or vice-versa. In both cases, the vertex with smallest range can be removed from the path, henceforth proving the existence of a shorter path: contradiction.
- If x_{i+1} is a left neighbor of x_i , then $\theta_l(x_i) \in I(x_{i-1}) \cap I(x_{i+1})$, proving that x_{i-1} and x_{i+1} are adjacent. By the way the path P can be shortened to $[x = x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_k = y]$: contradiction.

□

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

$$\begin{aligned}
 I(x_i^1) &= \begin{cases} [\theta_l(x_i), \theta_r(x_i)] & \text{if } \theta_l(x_i) \leq \theta_r(x_i) \\ [\theta_l(x_i), \theta_r(x_i) + 2\pi] & \text{if } \theta_l(x_i) > \theta_r(x_i) \end{cases} \\
 I(x_i^2) &= \begin{cases} [\theta_l(x_i) + 2\pi, \theta_r(x_i) + 2\pi] & \text{if } \theta_l(x_i) \leq \theta_r(x_i) \\ [\theta_l(x_i) + 2\pi, \theta_r(x_i) + 4\pi] & \text{if } \theta_l(x_i) > \theta_r(x_i) \end{cases}
 \end{aligned}$$

Intuitively, \tilde{G} is obtained from G by listing all the intervals according to increasing angle θ_l , starting from the arc of x_1 , and clockwise turning two rounds. So, each vertex x_i in G appears twice in \tilde{G} as x_i^1 and x_i^2 . It is straightforward to check that \tilde{G} is an interval graph.

Lemma 3.5 *For every $i < j$, $\text{dist}_G(x_i, x_j) = \min \{ \text{dist}_{\tilde{G}}(x_i^1, x_j^1), \text{dist}_{\tilde{G}}(x_j^1, x_i^2) \}$.*

Proof. Let $P = [x_i, x_{t_1}, \dots, x_{t_k}, x_j]$ be a shortest path between x_i and x_j in G . From Lemma 3.4 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 $i \leq t_1 \leq \dots \leq t_k \leq j$. Consider the path $P' = [x_i^1, x_{t_1}^1, \dots, x_{t_k}^1, x_j^1]$ of \tilde{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 $\text{dist}_G(x_i, x_j) \leq \text{dist}_{\tilde{G}}(x_i^1, x_j^1)$. We observe that any shortest path from x_i^1 to x_j^1 in \tilde{G} must be a right-path. By construction, every right-path of length l between a given pair of vertices in \tilde{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 \tilde{G} and $\text{dist}_G(x_i, x_j) = \text{dist}_{\tilde{G}}(x_i^1, x_j^1)$.

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 \tilde{G} between x_j^1 and x_i^2 . Indeed, since $i < j$, $\theta_l(x_i) \leq \theta_l(x_j) < \theta_l(x_i) + 2\pi$. Similarly to the previous case, P'' is a shortest path, proving that, in this case, $\text{dist}_G(x_i, x_j) = \text{dist}_{\tilde{G}}(x_j^1, x_i^2)$ and completing the proof of the lemma. \square

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 3.12 *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.*

4 Universal Distance Matrix

Given a family \mathcal{F} of graphs, [Chu90, AR02] considered the size of the smallest *induced universal graph* to give a measure of the density of \mathcal{F} . A graph $G_{\mathcal{U}}$ is an induced universal graph with respect to \mathcal{F} if any graph of \mathcal{F} is an induced subgraph of $G_{\mathcal{U}}$. In this section, we similarly define the *universal distance matrix* of a family \mathcal{F} , which 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 $B = (b_{i,j})_{1 \leq i,j \leq p}$ is an induced sub-matrix of a matrix $A = (a_{i,j})_{1 \leq i,j \leq q}$ if

there exists a sequence (s_1, \dots, s_p) of indices such that $b_{i,j} = a_{s_i, s_j}$, for all $i, j \in \{1, \dots, p\}$. Intuitively, a small universal distance matrix for \mathcal{F} indicates that many distance patterns repeat in many different graphs of \mathcal{F} .

Proposition 4.1 will be a tool for proving lower and upper bounds on the dimension of universal distance matrix. The proof technique is similar to the proof given by [KNR92] for relationship between adjacency schemes and *induced universal graphs* [AR02, Chu90].

Proposition 4.1 *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. 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 smaller than the largest integer contained in a subset S_G , which is the number of binary strings of length at most $\ell(n)$, i.e., $2^{\ell(n)+1} - 1$.

Now, assume 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 searching in M (which only depends on \mathcal{F}_n) the entry $M_{s^{-1}(\lambda), s^{-1}(\lambda')}$. As the two labels are extracted from the same graph, this value is the distance between the corresponding vertices. \square

From Proposition 4.1 and previously known bounds on distance labeling scheme [GPPR01], 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 $2^{\Omega(\sqrt{n})}$. By the way the dimension of the smallest universal distance matrix can be considered as a complexity measure of a graph family. From Proposition 4.1, Theorem 2.6 and 2.5 we obtain:

Corollary 4.2 *The dimension of the smallest universal distance matrix of n -vertex proper interval graphs is at most $\mathcal{O}(n^2)$ and at least $\Omega(n^2 / \log^2 n)$.*

Corollary 4.3 *The dimension of the smallest universal distance matrix of n -vertex interval graphs is at most $\mathcal{O}(n^5)$ and at least $\Omega(n^3/\log^4 n)$.*

References

- [AR02] S. Alstrup and T. Rauhe. Improved labeling scheme for ancestor queries. In *13th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 947–953, 2002.
- [BG05a] F. Bazzaro and C. Gavaille. Distance labeling for permutation graphs. In Elsevier, editor, *7th International Conference on Graph Theory (ICGT)*. *Electronical Notes in Discrete Mathematics*, September 2005.
- [BG05b] F. Bazzaro and C. Gavaille. Localized and compact data-structure for comparability graphs. Research Report RR-1343-05, LaBRI, University of Bordeaux 1, 351, cours de la Libération, 33405 Talence Cedex, France, February 2005.
- [BLS99] A. Brandstädt, V.B. Le, and J. Spinrad. *Graph classes, a survey*. Monographs on Discrete Mathematics and Applications. SIAM, 1999.
- [BV94] O. Berkman and U. Vishkin. Finding level-ancestors in trees. *Journal of Computer and System Sciences*, 48(2):214–230, 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.
- [CV03] B. Courcelle and R. Vanicat. Query efficient implementation of graphs of bounded clique width. *Discrete Applied Mathematics*, 131(1):129–150, 2003.
- [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.

- [GKK⁺01] C. Gavoille, M. Katz, N.A. Katz, C. Paul, and D. Peleg. Approximate distance labeling schemes. In *9th Annual European Symposium on Algorithms (ESA)*, number 2161 in Lecture Notes in Computer Science, pages 476–488, 2001.
- [GKR05] A. Gupta, A. Kumar, and R. Rastogi. Traveling with a pez dispenser (or, routing issues in mpl). *SIAM Journal on Computing*, 34(2):453–474, 2005.
- [GP03a] C. Gavoille and C. Paul. Distance labeling scheme and split decomposition. *Discrete Mathematics*, 273(1-3):115,130, 2003.
- [GP03b] C. Gavoille and D. Peleg. Compact and localized distributed data structures. *Journal of Distributed Computing*, 16:111,120, 2003.
- [GPPR01] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance labeling in graphs. In *12th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 210–219, 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*, number 450 in Lecture Notes in Computer Science, pages 101–108, 1990.
- [KKP00] M. Katz, N.A. Katz, and D. Peleg. Distance labeling schemes for well-separated graph classes. In *17th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, number 1770 in Lecture Notes in Computer Science, pages 516–528, 2000.
- [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 *19th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, number 2285 in Lecture Notes in Computer Science, pages 76–87, 2002.
- [KPR04] A. Korman, D. Peleg, and Y. Rodeh. Labeling schemes for dynamic tree networks. *Theory of Computing Systems*, 37:49–75, 2004.
- [MHP05] M. Mendel and S. Har-Peled. Fast construction of nets in low dimensional metrics, and their applications. In *21st Annual ACM Symposium on Computational Geometry (SoCG)*, June 2005.
- [Pel00a] D. Peleg. Informative labeling schemes for graphs. In *25th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, number 1893 in Lecture Notes in Computer Science, pages 579–588, 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.
- [Sli05] A. Slivkins. Distance estimation and object location via rings of neighbors. In *24th Annual ACM Symposium on Principles of Distributed Computing (PODC)*. ACM Press, July 2005.
- [Spi03] J. Spinrad. *Efficient graph representation*. Number 19 in Fields Institute Monographs. 2003.
- [Tal04] K. Talwar. Bypassing the embedding: Algorithms for low dimensional metrics. In *36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 281–290, June 2004.
- [Tho01] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. In *42th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 242–251, 2001.
- [Tho04] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6), November 2004.
- [TZ01] M. Thorup and U. Zwick. Approximate distance oracles. In *33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 183–192, 2001.
- [TZ05] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM*, 52(1), January 2005.
- [Weg67] G. Wegner. *Eigenschaften der Neven homologisch-einfacher Familien im R^n* . PhD thesis, University of Göttingen, 1967.