

OPTIMAL DISTANCE LABELING FOR INTERVAL GRAPHS AND RELATED GRAPH FAMILIES*

CYRIL GAVOILLE[†] AND CHRISTOPHE PAUL[‡]

Abstract. A distance labeling scheme is a distributed graph representation that assigns labels to the vertices and enables answering 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 [M. Katz, N. A. Katz, and D. Peleg, *Distance labeling schemes for well-separated graph classes*, in Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Comput. Sci. 1770, Springer-Verlag, Berlin, 2000, pp. 516–528]. Moreover, the scheme supports constant time distance queries, and if the interval representation of the input graph is given and the intervals are sorted, 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 nontrivial hereditary family with $2^{\Omega(n f(n))}$ unlabeled elements and with a distance labeling scheme with $f(n)$ bit labels.

Key words. graph representation, distance in graphs, distributed data-structure

AMS subject classifications. 05C62, 05C12, 68P30

DOI. 10.1137/050635006

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., [30] for an overview). Among the classical type of queries, let us mention those arising from the field of communication networks and distributed computing, like routing, connectivity, and distance queries. Peleg [26] introduced the concept of *informative labeling schemes* which generalizes the *implicit graph representation* proposed in [19]. 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 [14] for a survey).

This paper focuses on the *distance labeling scheme* [27], 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 with the vertex v in the graph G , and
- for all $x, y \in V$, $f(L(x, G), L(y, G)) = \text{dist}_G(x, y)$, 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 the distance labeling

*Received by the editors July 2, 2005; accepted for publication (in revised form) April 14, 2008; published electronically July 25, 2008. This work has been supported by the French research grant ANR-06-BLAN-0148 “Graph Decomposition and Algorithm - GRAAL.” An extended abstract of this work appears in *Proceedings of 11th Annual European Symposium on Algorithms (ESA '03)*, Lecture Notes in Comput. Sci. 2832.

<http://www.siam.org/journals/sidma/22-3/63500.html>

[†]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, Université Montpellier II, France (paul@lirmm.fr).

scheme [15] show that the family of n -vertex graphs requires $\Theta(n)$ bit labels while n -vertex trees need only $\Theta(\log^2 n)$ bit labels.¹ The case of dynamic tree networks has been studied in [23, 22, 24, 21]. The variant of approximated distance labeling schemes has been considered in [12, 32, 33, 34, 35, 29, 16, 25, 31]. 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 graphs, which is not tight with respect to the $\Omega(n^{1/3})$ lower bound [15]. For bounded degree graphs, $\Omega(\sqrt{n})$ is a lower bound on the label length. Finally, $\mathcal{O}(\log^2 n)$ -distance labeling schemes are known for interval and permutation graphs [20], for distance hereditary graphs [13], for bounded tree-width graphs, and, more generally, for bounded clique-width graphs [9].

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 [19], an 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 [19] asks whether any hereditary family with $2^{nf(n)}$ graphs of n vertices enjoys an $\mathcal{O}(f(n))$ -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 scheme.² It should be noticed that among the graph families listed above, the $\Omega(\log^2 n)$ lower bound on label length for n -vertex trees applies to neither interval nor permutation graphs. So the second question is: (2) *Can we improve the $\mathcal{O}(\log^2 n)$ upper bound [20] for either interval or permutation graphs?* This paper positively answers the above questions by considering the family of interval graphs. Interestingly, using some ideas from this paper, [2] positively answers the case of permutation graphs.

A graph $G = (V, E)$ is an *interval graph* (see, e.g., [4]) 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 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 interval defines an $\mathcal{O}(\log n)$ -adjacency labeling scheme, as it can be shown that all the interval 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 realizer; i.e., none of the intervals of the realizer is contained in another interval.

We show that any $\ell(n)$ -distance labeling on the n -vertex interval graph family requires labels of length $\ell(n) \geq 3 \log n - \mathcal{O}(\log \log n)$ bits. This lower bound derives from a new estimator on the number of interval graphs. We also obtain an asymptotically 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 $2 \lceil \log n \rceil$ -distance labeling scheme for proper interval graphs. We also prove that the bound for proper interval graphs is asymptotically 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 interestingly, if the list of intervals of the graph is

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.

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 remarking that any $\ell(n)$ -distance labeling scheme on a family \mathcal{F} converts trivially to a nondistributed data-structure for \mathcal{F} of $\mathcal{O}(\ell(n)n/\log n)$ space. The time complexity of distance queries remains unchanged, assuming 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 [5]. However, both approaches differ in essence. The technique of [5] 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 a query is mainly based on the efficient parallel implementation of *level ancestor queries* on trees (to compute z) of [3]. However, this clever scheme cannot be converted into a distributed data-structure as ours can be 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 [15], this cannot be done in less than $\Omega(\log^2 n)$ bit labels. So, access to a global data-structure is inherent in the approach of [5].

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 asymptotically 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 an α -linkable subfamily (introduced in subsection 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 labeled interval graphs with n vertices. Computing $I(n)$ is an unsolved graph-enumeration problem. Cohen, Komlós, and Muller [7] gave the probability $p(n, m)$ that a labeled n -vertex m -edge random graph is an interval graph under conditions on m . They computed $p(n, m)$ for $m < 4$ and showed, more generally, that

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

As the total number of labeled n -vertex m -edge graphs is $\binom{n}{m}$, a formula of $p(n, m) \cdot \binom{n}{m}$ follows for the number of labeled interval graphs with $m = \Theta(n^{5/6})$ edges. Unfortunately, using this formula results in $I(n) \geq 2^{\Omega(n^{5/6} \log n)} = 2^{o(n)}$, which is a

lower bound too weak for our needs. The exact number of interval graphs is given up to 30 vertices in [17]. Actually, the generating functions for interval and proper interval graphs (labeled and unlabeled) are known [17], 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 [17] left open 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 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 1. *The number $I(n)$ of labeled n -vertex 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 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 that $|S_i| = k$ 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 is 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 $\mathcal{I}(b_0) = [0, k]$; and
- $\mathcal{I}(c_j) = [p(k+1) + j, p(k+1) + j + k]$.

As depicted in Figure 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

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

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

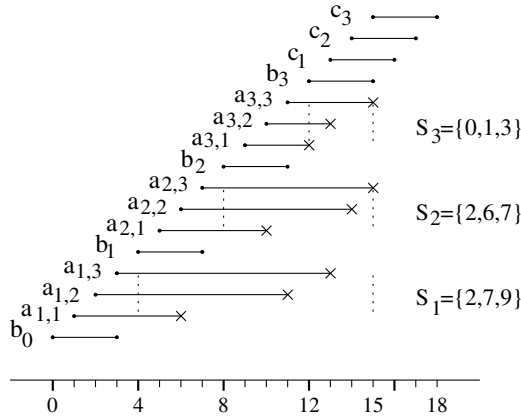


FIG. 1. The interval representation of G_S for $S = (S_1, S_2, S_3)$.

By construction $\mathcal{F}_{p,k}$ contains at most $|\mathcal{S}_{p,k}|$ unlabeled graphs, and each one has at most $(|A| + |B \setminus \{b_0\}|)!$ 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 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 give 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 also allows the recovery of 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 equal to 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 come 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 assumption b_{i-1} and L_{i-1} are identified, so the above properties also hold for the graph $(G \setminus L_{i-1}) \cup \{b_{i-1}\}$. Combining properties 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 (applying properties 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).

Recovering 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 equation (1).

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

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

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

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

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

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

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 1. \square

Each vertex can be described by its interval representation, and since there are at most $2n$ interval boundaries ranging in $[1, 2n]$, adjacency labels of length $\lceil 2 \log(2n) \rceil$ suffice. Concatenating such labels according to the vertex ordering of the graph yields an upper bound on the number of labeled n -vertex graphs, i.e., $I(n) \leq \lceil 2 \log(2n) \rceil^n \leq 2n \log n + \mathcal{O}(n)$. It follows that $\frac{1}{n} \log I(n) \leq 2 \log n + \mathcal{O}(1)$, and so the lower bound of Theorem 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 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 $l = r$, each one of eccentricity at most α . 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 nonnull 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 Figure 2).

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 nonnull 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 subfamily. As we

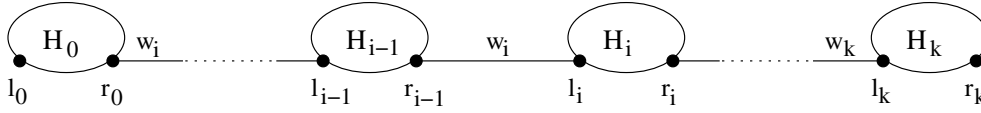


FIG. 2. Linking a sequence of α -graphs.

shall see, interval graphs support a large 1-linkable subfamily (we mean large with respect to n), whereas proper interval graphs support a large 2-linkable subfamily.

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

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 subfamily \mathcal{H} . Each graph H_i has $\nu = \Theta(n/\log n)$ vertices, and consecutive H_i 's from \mathcal{S} are spaced with paths of length $\Theta(n/\log n)$. Intuitively, the term $\frac{1}{\nu} \log H(\nu)$ measures the minimum label length required to compute the distance between any two vertices within the same H_i 's (i.e., whether they are adjacent or at distance at least two). The $\log \nu$ term is required to compute the distance between vertices of distinct H_i 's. The difficulty is to show that some vertices require both pieces of information, observing that one can distribute information on the vertices in a nontrivial way. For instance, the two extreme vertices of a path of length w_i do 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 1. Let p, q, k be nonnull 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, $|\mathcal{W}| = q^k$. Let \mathcal{H} be an arbitrary α -linkable subfamily 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 Figure 2).

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

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 taking a vertex label successively in H_0, H_1, \dots, H_k . More formally,

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

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

CLAIM 1. *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 length 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 $t \geq \log(|\mathcal{L}|^{1/(k+1)} + 1) - 1 \geq \frac{1}{k+1} \log |\mathcal{L}| - 1$.

CLAIM 2. $|\mathcal{L}_W|^p \geq H(p)^{k+1}$. 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_S of $(k + 1)$ -tuples can be associated with each \mathcal{S}^W ,

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

each $(k + 1)$ -tuple being composed by the labels of an arbitrary vertex in each H_i . We have $\bigcup_{\mathcal{S}^W \in \mathcal{H}_p(W)} L_S \subseteq (\mathcal{L}_W)^p$. Let us show that $L_S \neq L_{S'}$ 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_S and $L_{S'}$, $L_S \neq L_{S'}$ follows. It implies that

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

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, Claim 2 follows: we have $|\mathcal{L}_W|^p \geq H(p)^{k+1}$.

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

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

Let us now show that interval graphs and proper interval graphs both enjoy a large enough α -linkable subfamily of labeled graphs. The lower bounds on distance labeling schemes for these two graph families therefore follow from Lemma 1 and Theorem 1.

THEOREM 2. *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 $I(\nu)$ be the number of labeled interval graphs with ν vertices. Let \mathcal{H} be that family of labeled ν -vertex interval graphs enjoying a universal vertex (adjacent to any other vertex) where the universal vertex is labeled ν . Any graph of \mathcal{H} is a 1-graph (set the universal vertex as $l = r$). And note that \mathcal{H} is a 1-linkable subfamily of the family of interval graphs (see Figure 3). Moreover, it is straightforward to see that any $(\nu - 1)$ -vertex labeled interval graph is an induced subgraph of some $H \in \mathcal{H}$. It follows that the number $H(\nu)$ of ν -vertex labeled graphs of \mathcal{H} is precisely $I(\nu - 1)$.

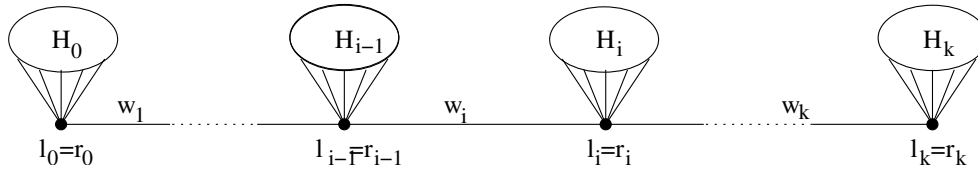


FIG. 3. Each H_i is a 1-linkable ν -interval graph with a universal vertex $l_i = r_i$.

By Lemma 1 applied to \mathcal{H} and the interval graph family, the maximum length t of a label of length is at least

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

From Theorem 1, we have

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

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

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

Gathering everything together, we obtain

$$\begin{aligned} t &\geq 2 \log n - 3 \log \log n + \log \log \log n + \log \lfloor n / \log n \rfloor - \mathcal{O}(1) \\ &\geq 3 \log n - 4 \log \log n, \end{aligned}$$

completing the proof. \square

THEOREM 3. *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 Q_p be the intersection graph of the interval family $\{\mathcal{I}(x_i) = [i, i + p], \mathcal{I}(y_i) = [i + p, i + 2p - 1], 1 \leq i \leq p\}$. (See Figure 4.) Since any interval is of length p , Q_p is a proper interval graph with $2p$ vertices.

Let \mathcal{H} be the family of labeled graphs isomorphic to Q_p such that the vertices x_1 and y_1 are labeled 1 and 2, respectively. Let $Q \in \mathcal{H}$. It is not difficult to see by

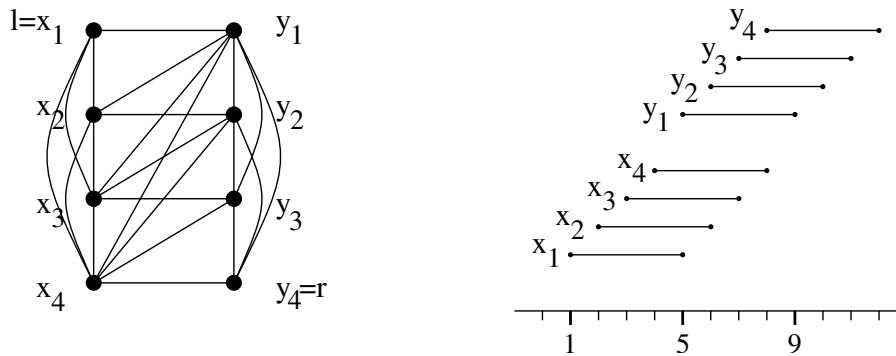


FIG. 4. The graph Q_4 and its interval representation.

induction that once x_i and y_i have been identified in Q , 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 $Q_{p-(i-1)}$), and observe that in this latter graph (1) x_{i+1} is the neighbor of x_i , distinct from y_i , with minimum degree; and (2) y_{i+1} is the only nonneighbor of x_i that is adjacent to x_{i+1} . Therefore, the cardinality of \mathcal{H} is $(2p - 2)!$.

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.

Let $\nu = 2p$. The cardinality $H(\nu)$ of \mathcal{H} is $(\nu - 2)!$. We note that $\frac{1}{\nu} \log(\nu - 2)! \geq \log \nu - \mathcal{O}(1)$. By Lemma 1, for $\alpha = 2$ and $\nu = \lfloor n/(2 \log n) \rfloor$, the length of the maximum label for n -vertex proper interval graphs is at least

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

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, the input interval graph G is supposed to be an n -vertex connected graph, and \mathcal{I} is given as a realizer. We then denote by $\mathcal{I}(x)$ the interval associated with vertex x and by $L(x)$ and $R(x)$, respectively, the left and the right boundaries of $\mathcal{I}(x)$. As in [5], the intervals of \mathcal{I} are assumed to be sorted according to the left boundaries. We will also assume w.l.o.g. that the left boundaries are pairwise distinct and range in $\{1, \dots, 2n\}$. These hypotheses are not restrictive since in $\mathcal{O}(n)$ time one can scan all the intervals and their boundaries from $\min_x L(x)$ to $\max_x L(x)$ and compute another realizer for G with sorted and distinct boundaries in $\{1, \dots, 2n\}$.

3.1. Reduction to proper interval graphs. Proper interval graphs are interval graphs having an inclusion-free realizer. There are several recognition algorithms that run in linear time, that is, $\mathcal{O}(n + m)$ time, where n is the number of vertices and m the number of edges; cf. [8, 10, 18]. Moreover, these algorithms output an inclusion-free realizer.

It is well known that the family of proper interval graphs can be characterized as the family of $K_{1,3}$ -free³ interval graphs [37] (see Figure 5) or as the family of intersection graphs of a family of unit length intervals [28].



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

Reducing the interval graph to a proper interval graph, the interval realizer we obtain yields a slightly different characterization of proper interval graphs. Such a realizer is called a *proper realizer*.

DEFINITION 1. A realizer \mathcal{I} of an interval graph is proper if for every pair of intervals $\mathcal{I}(x)$ and $\mathcal{I}(y)$ such that $\mathcal{I}(x) \subseteq \mathcal{I}(y)$, $R(x) = R(y)$.

Let us notice that an inclusion-free realizer of intervals is a proper realizer; indeed, only special inclusions are allowed in a proper realizer. The reverse is also true: any interval graph having a proper realizer is $K_{1,3}$, i.e., is a proper interval graph. Let \mathcal{I} be a proper realizer of an interval graph G . Assume by contradiction that G contains a set of four vertices that induces a $K_{1,3}$; say, a, b, c are three independent vertices adjacent to a common neighbor x . W.l.o.g. assume that $L(a) < R(a) < L(b) < R(b) < L(c) < R(c)$. It follows that $L(x) \leq R(a)$ and $L(c) \leq R(x)$. Thereby we have $\mathcal{I}(b) \subset \mathcal{I}(x)$ and $R(b) \neq R(x)$, which is a contradiction; the realizer \mathcal{I} is supposed to be proper.

Our reduction is based on the notion of an *enclosing neighbor*.

DEFINITION 2. Let \mathcal{I} be a nonproper 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 . The enclosing neighbor of x , denoted by $N_e(x)$, is a neighbor of x for which $\mathcal{I}(x) \subset \mathcal{I}(N_e(x))$ such that the following conditions are satisfied:

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

As the left boundaries of \mathcal{I} are pairwise disjoint, the enclosing neighbor, if it exists, is unique for every vertex. From a realizer \mathcal{I} , we can construct a proper realizer \mathcal{I}' 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 set $\mathcal{I}'(x) = \mathcal{I}(x)$ otherwise.

Let us remark that

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 $G' = (V, E' \subseteq E)$, the graph realized by \mathcal{I}' .

We say that a vertex x has been *extended* in G' if $R'(x) > R(x)$.

LEMMA 2. The realizer \mathcal{I}' is proper. It can be computed from \mathcal{I} in $\mathcal{O}(n)$ time.

³That is, the graph does not contain a $K_{1,3}$ graph as an induced subgraph.

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 that $\mathcal{I}'(v)$ is maximal for 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 . As $R(u) \leq R'(u)$, we have $L(v) < L(u) < R(u) < R'(v)$. Now, $R(v) = R'(v)$ implies $\mathcal{I}(u) \subset \mathcal{I}(v)$, meaning that u has an enclosing neighbor. It follows that $R'(u) \geq R(v) = R'(v)$, contradicting $R'(u) < R'(v)$. So we proved that for any pair of vertices u and v such that $\mathcal{I}'(u) \subset \mathcal{I}'(v)$, we have $R'(u) = R'(v)$; thereby \mathcal{I}' is proper.

The realizer \mathcal{I}' can be computed as follows: scan the intervals of \mathcal{I} according 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 $\{x \mid L(x_i) < L(x) \text{ and } R(x_i) < R(x)\}$. Now for any vertex $v \notin S$, its enclosed neighbor is the maximum $x_i \in S$ such that $L(x_i) < L(v)$. Two scans of the vertices suffice to compute those enclosed neighbors. As by construction the vertices of S are maximal for the inclusion, they do not have enclosed neighbors. It follows that computing \mathcal{I}' requires only $\mathcal{O}(n)$ time. \square

The main result of this subsection shows that distances in G can be retrieved from the distances in G' whose realizer \mathcal{I}' is proper. To prove Theorem 4, the lemma below is required.

LEMMA 3. *If x and y are two vertices such that $R(x) = R'(x) \leq R(y)$, then $\text{dist}_G(x, y) = \text{dist}_{G'}(x, y)$.*

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 $N_e(u_j)$ yields a shortest x, y -path P' in G' using fewer extended vertices than P , which is a contradiction; P does not contain any extended vertex. \square

DEFINITION 3. *For any vertex x , the maximum neighbor of x , denoted by $N_m(x)$, is a neighbor of x such that $R(N_m(x))$ is maximum.*

Like enclosing neighbors, maximum neighbors are not extended in G' ; i.e., for any x , $R'(N_m(x)) = R(N_m(x))$. 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 $v = N_m(x)$, which is a contradiction.

Let us define $\text{adj}_G(x, y)$ as 1 if x and y are adjacent in G and as 0 otherwise.

THEOREM 4. *Let x and y be two vertices such that $R(x) \leq R(y)$. Then,*

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

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, $\text{dist}_{G'}(N_m(x), y) = \text{dist}_G(N_m(x), y)$. 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 $k - 1$, which is a contradiction. \square

Given a distance labeling scheme of a proper interval graph, (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 $\text{adj}_G(x, y)$; and
- $L'(x, G')$ 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 1. *If there exists a distance labeling scheme for n -vertex proper interval graphs using labels of length $s(n)$, then the n -vertex interval graph family enjoys a distance labeling scheme using labels of length $2 \lceil \log(2n) \rceil + 2s(n)$.*

3.2. Distance labeling scheme of proper interval graphs. Let us now assume that a proper realizer \mathcal{I} for a graph $G = (V, E)$ is given (so G is a proper interval graph). Let x_0 be the vertex with the minimum left boundary. Notice that x_0 also has a minimum right boundary. We define the *layer partition* V_0, V_1, \dots, V_k (see Figure 6) as the partition of vertices into distance layers from x_0 :

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

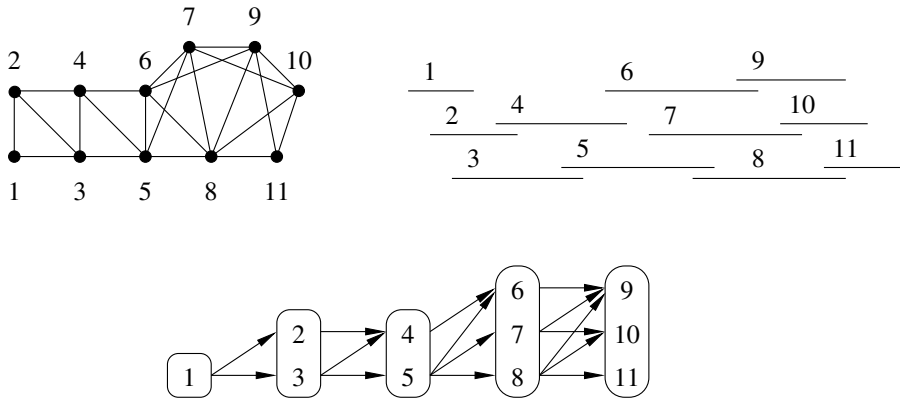


FIG. 6. A proper interval graph with an interval representation and the associated layer partition.

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 4. *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 5. For all distinct vertices x and y such that $\lambda(x) \leq \lambda(y)$,

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

To prove Theorem 5, we need the following preliminary result.

LEMMA 4. For all $i \geq 1$, let x_{i-1} be the vertex of V_{i-1} having maximum right boundary.⁴ Then $\{x_{i-1}\} \cup V_i$ is a clique of G .

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. This 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 $R(x_{i-1})$. This implies that V_i is a clique. \square

Proof of Theorem 5. 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 4 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 4, 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 $\text{adj}_{H^t}(x, y) = 1$. This 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 6. There exists a total ordering π of the vertices such that

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

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

Proof. We first claim that such an ordering π is obtained by an elimination ordering of the vertices such that at each step, the sink of the subgraph of H induced by the noneliminated vertices that has a minimum left bound is removed. This can be implemented by using a depth-first search of H : the ordering π is the postorder traversal of the search (i.e., the order in which the vertices have been removed from the LIFO-stack).

In the example in Figure 6, we obtain the following 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 $L(x) < L(y)$).

⁴For $i > 0$, x_i is the maximum neighbor of x_{i-1} and $x_0 \in V_0$.

CLAIM 5. Let z be a vertex such that $\lambda(y) < \lambda(z)$. If $\text{adj}_{H^t}(x, z) = 1$, then $\text{adj}_{H^t}(y, z) = 1$.

Since $\text{adj}_{H^t}(x, z) = 1$, the graph G contains a x, z -shortest path of length $\lambda(z) - \lambda(x)$. Let $P = [x, \dots, v_{\lambda(y)-1}, v_{\lambda(y)}, v_{\lambda(y)+1}, \dots, z]$ 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 $z = v_{\lambda(y)+1}$. Note that $\text{adj}_{H^t}(x, y) = 0$ implies that $\text{adj}_{H^t}(v_{\lambda(y)-1}, y) = 0$. It follows that $v_{\lambda(y)-1}$ and y are nonadjacent in G . Now if $\text{adj}_{H^t}(y, z) = 0$, it similarly implies that $\text{adj}_{H^t}(y, v_{\lambda(y)+1}) = 0$, and thus y and $v_{\lambda(y)+1}$ are also nonadjacent vertices of G . As P is a shortest path, $v_{\lambda(y)-1}$ and $v_{\lambda(y)+1}$ are nonadjacent in G . As by construction $\lambda(v_{\lambda(y)}) = \lambda(y)$, y is adjacent in G to $\lambda(v_{\lambda(y)})$. It follows that in G the set of vertices $\{y, v_{\lambda(y)-1}, v_{\lambda(y)}, v_{\lambda(y)+1}\}$ induce a $K_{1,3}$, which is a contradiction; G is a proper interval graph. Claim 5 follows: if $\text{adj}_{H^t}(x, z) = 1$, then $\text{adj}_{H^t}(y, z) = 1$.

Note also that at the step when 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 $\pi(x) < \pi(y)$, and we are done. So assume x is not a sink. Therefore, there exists z such that $\text{adj}_{H^t}(x, z) = 1$. By Claim 5, 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 the 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 eliminated before y . Recursively applying this argument enables us to prove that x will be eliminated before y by the algorithm, and so $\pi(x) < \pi(y)$.

Let us now analyze 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 is 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

In passing, let us point out an interesting combinatorial property of poset H^t . The *dimension* of a poset P on a set V is the minimum number d of total orders ρ_1, \dots, ρ_d on V such that $x <_P y$ iff $x <_{\rho_i} y$ for every i [11, 36]. By choosing $\rho_1 = \pi$ and ρ_2 as any linear extension of λ , Theorem 6 immediately implies the following corollary.

COROLLARY 2. 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 (3) and (4). Clearly, $\lambda(x), \pi(x) \in \{1, \dots, n\}$, so the main result follows.

THEOREM 7. 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 1 and Theorem 7, 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 $\mathsf{R}(x)$ and $\mathsf{L}(x)$, the boundaries of the interval $\mathcal{I}(x)$, plus the labels of x and $\mathsf{N}_m(x)$ in the extended proper interval graph G' .

The improvement is based on the observation that $\lambda(x)$ and $\lambda(\mathsf{N}_m(x))$ differ at most by one ($\lambda(\mathsf{N}_m(x)) - \lambda(x) \in \{0, 1\}$). Indeed, there is no edge in G between a pair of vertices that does not belong to the same distance layer or consecutive distance layers. Therefore, a single bit, say, $b(x)$, suffices to recover $\lambda(\mathsf{N}_m(x))$ from $\lambda(x)$. Therefore, the label of a vertex x can be defined as follows:

$$L(x, G) = \langle \mathsf{L}(x), \mathsf{R}(x), \lambda(x), \pi(x), b(x), \pi(\mathsf{N}_m(x)) \rangle.$$

As $1 \leq \mathsf{L}(x) < \mathsf{R}(x) \leq 2n$, 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 $5 \lceil \log n \rceil + 3$. Thus, we have the following theorem.

THEOREM 8. *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.*

Proof. We have already discussed the length of the labels. The distance decoder takes a constant time. Concerning the time complexity of constructing all the labels, let us first observe that in $\mathcal{O}(n)$ time one can compute $\mathsf{N}_m(x)$ for each x as follows: Since \mathcal{I}' is proper, a maximum neighbor of a given vertex x is precisely the vertex y with maximum left boundary satisfying $\mathsf{L}(y) \leq \mathsf{R}(x)$. Since we made the assumption that the intervals have been given sorted with respect to their left boundaries, a single scan enables us to compute the maximum neighbor of each vertex.

Since the distance labels in proper interval graphs can be computed in $\mathcal{O}(n)$ time, and since G' can be obtained within the same complexity, all the labels of G can be computed in $\mathcal{O}(n)$ time. \square

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 with 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 in Figure 7.

A neighbor y of x is a *right neighbor* (resp., *left neighbor*) 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 be neither 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 5. *Every shortest path between nonadjacent vertices x and y is either a right-path from x to y or a right-path from y to x .*

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 that 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 a right-path, there is an integer i , $1 \leq i < k$, such

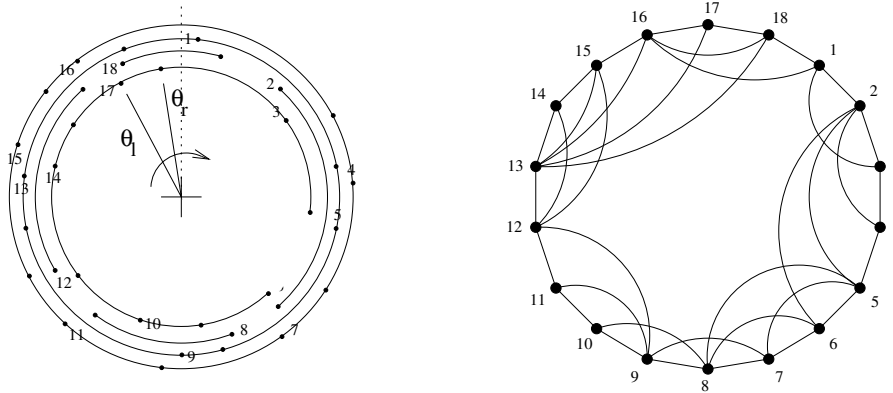


FIG. 7. A circular-arc graph with its arc representation. Vertices are labeled from 1 to 18 according to their value θ_l .

that x_{i+1} is not a right neighbor of x_i and such that the subpath 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 the smallest range can be removed from the path, henceforth proving the existence of a shorter path, which is a 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 $[x = x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_k = y]$, which is a contradiction. \square

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 $1 \leq i \leq n$,

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

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

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 turning clockwise around twice. 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 6. 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 5 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 a circular-arc graph family by doubling the number of vertices and the label length.

THEOREM 9. *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, [6, 1] considered the size of the smallest *induced universal graph* to give a measure of the density of \mathcal{F} . A graph \mathcal{G} is an induced universal graph with respect to \mathcal{F} if any graph of \mathcal{F} is an induced subgraph of \mathcal{G} . 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 an *induced submatrix*. A matrix $B = (B_{i,j})_{1 \leq i,j \leq p}$ is an induced submatrix 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 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 [19] for the relationship between adjacency schemes and induced universal graphs [1, 6].

PROPOSITION 1. *If \mathcal{F} enjoys an ℓ -distance labeling scheme, then \mathcal{F} has a universal distance matrix of dimension $2^{\ell+1} - 1$. If \mathcal{F} has a universal distance matrix of dimension d , then \mathcal{F} enjoys a $\lfloor \log d \rfloor$ -distance labeling scheme.*

Proof. For every integer $i \geq 1$, let $\phi(i)$ denote the i th binary string in the lexicographic order $\phi(1) = \epsilon$, $\phi(2) = 0$, $\phi(3) = 1$, $\phi(4) = 00$, $\phi(5) = 01$, $\phi(6) = 10$, $\phi(7) = 11$, $\phi(8) = 000, \dots$. We note that the length of $\phi(i)$ is $\lfloor \log i \rfloor$. We denote by ϕ^{-1} the inverse function of ϕ .

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: $M_{i,j} = f(\phi(i), \phi(j))$ if there exists a graph $G \in \mathcal{F}$ with two vertices u and v such that $L(u, G) = \phi(i)$ and $L(v, G) = \phi(j)$, and $M_{i,j} = \infty$ otherwise.

With every $G \in \mathcal{F}$, one can associate the subset $I_G = \{\phi^{-1}(L(u, G)) \mid u \in V(G)\}$ of indices. By construction, for all $i, j \in I_G$, $M_{i,j} = f(\phi(i), \phi(j))$, which is the distance in G between the vertices labeled $\phi(i)$ and $\phi(j)$. So M contains as an induced submatrix the distance matrix of every $G \in \mathcal{F}$: M is universal. Moreover, its dimension is smaller than the largest integer contained in a subset I_G , which cannot exceed the number of binary strings of length at most ℓ , i.e., $2^{\ell+1} - 1$.

Now, assume M is a universal distance matrix of dimension d 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 submatrix of M with a sequence of indices I_G , then we label the vertex corresponding to the index $i \in I_G$ by the binary string $\phi(i)$. The length of

this label is the length of $\phi(i)$. The length is therefore bounded by the length of $\phi(d)$, that is, $\lfloor \log d \rfloor$ bits. Now, given two binary labels λ and λ' , extracted from the same graph, one can compute the distance by searching in M (which depends only on \mathcal{F}) the entry $M_{\phi^{-1}(\lambda), \phi^{-1}(\lambda')}$. As the two labels are extracted from the same graph, this value is the distance between the corresponding vertices. \square

From Proposition 1 and previously known bounds on the distance labeling scheme [15], 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 1 and Theorems 3 and 2 we obtain the following corollaries.

COROLLARY 3. *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. *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

- [1] S. ALSTRUP AND T. RAUHE, *Improved labeling scheme for ancestor queries*, in Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, SIAM, Philadelphia, 2002, pp. 947–953.
- [2] F. BAZZARO AND C. GAVOILLE, *Localized and compact data-structure for comparability graphs*, Discrete Math., (2008), to appear.
- [3] O. BERKMAN AND U. VISHKIN, *Finding level-ancestors in trees*, J. Comput. System Sci., 48 (1994), pp. 214–230.
- [4] A. BRANDSTÄDT, V.B. LE, AND J. SPINRAD, *Graph Classes, A Survey*, SIAM Monogr. Discrete Math. Appl. 3, SIAM, Philadelphia, 1999.
- [5] D.Z. CHEN, D.T. LEE, R. SRIDHAR, AND C.N. SEKCHARAN, *Solving the all-pair shortest path query problem on interval and circular-arc graphs*, Networks, 31 (1998), pp. 249–257.
- [6] F.R.K. CHUNG, *Universal graphs and induced-universal graphs*, J. Graph Theory, 14 (1990), pp. 443–454.
- [7] J.E. COHEN, J. KOMLÓS, AND T. MUELLER, *The probability of an interval graph, and why it matters*, in Relations Between Combinatorics and Other Parts of Mathematics, Sympos. Pure Math. 34, AMS, Providence, RI, 1979, pp. 97–115.
- [8] D.G. CORNEIL, H. KIM, S. NATARAJAN, S. OLARIU, AND A.P. SPRAGUE, *Simple linear time algorithm of unit interval graphs*, Inform. Process. Lett., 55 (1995), pp. 99–104.
- [9] B. COURCELLE AND R. VANICAT, *Query efficient implementation of graphs of bounded clique width*, Discrete Appl. Math., 131 (2003), pp. 129–150.
- [10] C.M. HERRERA DE FIGUEIREDO, J. MEIDANIS, AND C. PICININ DE MELLO, *A linear-time algorithm for proper interval recognition*, Inform. Process. Lett., 56 (1995), pp. 179–184.
- [11] B. DUSHNIK AND E.W. MILLER, *Partially ordered sets*, Amer. J. Math., 63 (1941), pp. 600–610.
- [12] C. GAVOILLE, M. KATZ, N.A. KATZ, C. PAUL, AND D. PELEG, *Approximate distance labeling schemes*, in Proceedings of the 9th Annual European Symposium on Algorithms (ESA), Lecture Notes in Comput. Sci. 2161, Springer-Verlag, New York, 2001, pp. 476–488.
- [13] C. GAVOILLE AND C. PAUL, *Distance labeling scheme and split decomposition*, Discrete Math., 273 (2003), pp. 115–130.
- [14] C. GAVOILLE AND D. PELEG, *Compact and localized distributed data structures*, Distributed Computing, 16 (2003), pp. 111–120.
- [15] C. GAVOILLE, D. PELEG, S. PÉRENNES, AND R. RAZ, *Distance labeling in graphs*, in Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM, New York, SIAM, Philadelphia, 2001, pp. 210–219.
- [16] A. GUPTA, A. KUMAR, AND R. RASTOGI, *Traveling with a pez dispenser (or, routing issues in MPLS)*, SIAM J. Comput., 34 (2005), pp. 453–474.
- [17] P. HANLON, *Counting interval graphs*, Trans. Amer. Math. Soc., 272 (1982), pp. 383–426.
- [18] P. HELL, J. BANG-JENSEN, AND J. HUANG, *Local tournaments and proper circular arc graphs*, in Algorithms, Lecture Notes in Comput. Sci. 450, Springer-Verlag, Berlin, 1990, pp. 101–108.
- [19] S. KANNAN, M. NAOR, AND S. RUDICH, *Implicit representation of graphs*, SIAM J. Discrete Math., 5 (1992), pp. 596–603.

- [20] M. KATZ, N.A. KATZ, AND D. PELEG, *Distance labeling schemes for well-separated graph classes*, in Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Comput. Sci. 1770, Springer-Verlag, Berlin, 2000, pp. 516–528.
- [21] A. KORMAN, *General compact labeling schemes for dynamic trees*, in Proceedings of the 19th International Symposium on Distributed Computing (DISC), Lecture Notes in Comput. Sci. 3724, Springer-Verlag, Berlin, 2005, pp. 457–471.
- [22] A. KORMAN AND D. PELEG, *Labeling schemes for weighted dynamic tree*, in Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Comput. Sci. 2719, Springer-Verlag, Berlin, 2003, pp. 369–383.
- [23] A. KORMAN, D. PELEG, AND Y. RODEH, *Labeling schemes for dynamic tree networks*, in Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Comput. Sci. 2285, Springer-Verlag, Berlin, 2002, pp. 76–87.
- [24] A. KORMAN, D. PELEG, AND Y. RODEH, *Labeling schemes for dynamic tree networks*, Theory Comput. Syst., 37 (2004), pp. 49–75.
- [25] M. MENDEL AND S. HAR-PELED, *Fast construction of nets in low-dimensional metrics and their applications*, SIAM J. Comput., 35 (2006), pp. 1148–1184.
- [26] D. PELEG, *Informative labeling schemes for graphs*, in Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science (MFCS), Lecture Notes in Comput. Sci. 1893, Springer-Verlag, Berlin, 2000, pp. 579–588.
- [27] D. PELEG, *Proximity-preserving labeling schemes*, J. Graph Theory, 33 (2000), pp. 167–176.
- [28] F.S. ROBERTS, *Indifference graphs*, in Proof Techniques in Graph Theory, Academic Press, New York, 1969, pp. 139–146.
- [29] A. SLIVKINS, *Distance estimation and object location via rings of neighbors*, in Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 2005, pp. 41–50.
- [30] J. P. SPINRAD, *Efficient Graph Representations*, Fields Inst. Monogr. 19, AMS, Providence, RI, 2003.
- [31] K. TALWAR, *Bypassing the embedding: Algorithms for low dimensional metrics*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 2004, pp. 281–290.
- [32] M. THORUP, *Compact oracles for reachability and approximate distances in planar digraphs*, in Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, Los Alamitos, CA, 2001, pp. 242–251.
- [33] M. THORUP, *Compact oracles for reachability and approximate distances in planar digraphs*, J. ACM, 51 (2004), pp. 993–1024.
- [34] M. THORUP AND U. ZWICK, *Approximate distance oracles*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 2001, pp. 183–192.
- [35] M. THORUP AND U. ZWICK, *Approximate distance oracles*, J. ACM, 52 (2005), pp. 1–24.
- [36] W. T. TROTTER, *Combinatorics and Partially Ordered Sets: Dimension Theory*, The Johns Hopkins University Press, Baltimore, MD, 1992.
- [37] G. WEGNER, *Eigenschaften der Neven homologisch-einfacher Familien im R^n* , Ph.D. thesis, University of Göttingen, Göttingen, Germany, 1967.