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

Research Report RR-1250-00

Approximate Distance Labeling Schemes

by Cyril Gavoille, Michal Katz,
Nir A. Katz, Christophe Paul and David Peleg

December, 2000

Approximate Distance Labeling Schemes

Cyril Gavoille* Michal Katz† Nir A. Katz† Christophe Paul*
David Peleg‡

December 22, 2000

Abstract

We consider the problem of labeling the nodes of an n-node graph G with short labels in such a way that the distance between any two nodes u, v of G can be approximated efficiently (in constant time) by merely inspecting the labels of u and v, without using any other information. We develop such constant approximate distance labeling schemes for the classes of trees, bounded treewidth graphs, planar graphs, k-chordal graphs, and graphs with a dominating pair (including for instance interval, permutation, and AT-free graphs). We also show lower bounds, and prove that most of our schemes are optimal in length of labels generated and in the quality of the approximation, leaving some open problems.

*LaBRI, Université Bordeaux I, 351, cours de la Libération, 33405 Talence Cedex, France. {*gavoille,paul*}*@labri.u-bordeaux.fr*.

†Department of Applied Mathematics, Bar Ilan University, Ramat Gan, 52900, Israel. {*mkatz,katz*}*@isys-oms.com*.

‡Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, 76100 Israel. *peleg@wisdom.weizmann.ac.il*.

1 Introduction

1.1 Approximate Distance Labeling

Let us define the notion of approximate distance labeling schemes more precisely. Given a graph G (assumed undirected and connected) and two nodes u and v, let $d_G(u,v)$ denote the distance between u and v in G. A *node-labeling* for the graph G is a non-negative integer function L that assigns a label $L(u,G)$ to each node u of G.

A *distance decoder* is an integer function f responsible for distance computation; given two labels λ_1, λ_2 (not knowing from which graph they are taken), it returns $f(\lambda_1, \lambda_2)$. We say that the pair $\langle L, f\rangle$ is an *distance labeling* for G if $f(L(u,G), L(v,G)) = d_G(u,v)$ for any pair of nodes $u, v \in V(G)$. The distance labeling $\langle L, f\rangle$ is (s,r)*-approximate* if

$$d_G(u,v) \leqslant f(L(u,G), L(v,G)) \leqslant s \cdot d_G(u,v) + r \tag{1}$$

for any pair of nodes $u, v \in V(G)$. More generally, $\langle L, f\rangle$ is an (s,r)-approximate *distance labeling scheme* for the graph family $\mathcal{G}$ if it is an (s,r)-approximate distance labeling for every graph $G \in \mathcal{G}$. This paper concerns the existence of distance labeling schemes which use short labels. Let $|L(u,G)|$ denote the length of the binary label $L(u,G)$ associated with u in $G \in \mathcal{G}$, and denote

$$\ell_{\langle L,f\rangle}(\mathcal{G}) = \max_{G\in\mathcal{G}} \max_{u\in V(G)} |L(u,G)|$$

and

$$\ell_{(s,r)}(\mathcal{G}) = \min_{\langle L,f\rangle} \left\{ \ell_{\langle L,f\rangle}(\mathcal{G}) \right\}$$

where the minimum is taken over all (s,r)-approximate distance labeling schemes $\langle L, f\rangle$ for $\mathcal{G}$.

In this paper, we are also interested in the *query time*, i.e., the worst-case time complexity of the distance decoder. We assume a ω-bit word RAM model of computation. Each memory word can contain an integer in the range $[0, 2^\omega)$. The instruction set available consists of standard arithmetic, bitwise logical, and comparison operations on words, all assumed to require constant time. For the use of bucketing and perfect hashing functions, we assume that the arithmetic operation set contains integer multiplication and integer division. We assume that the input graph is of size n, and in order to access all the bits of the input, we require that $\omega \geqslant \log n$. However, integers larger than ω bits can be treated considering t-uple of words. For constant t, all the instruction set for $t\omega$-bit word RAM computer, i.e., a RAM computer with $O(\omega)$-bit words, are computable in constant time on a simple ω-bit word RAM computer. So we assume $\omega = \Omega(\log n)$.

We focus on two important special cases of (s,r)-approximate distance labeling schemes. An $(s,0)$-approximate distance labeling scheme is referred to as a *s-multiplicative* distance labeling scheme. Analogously, a $(1,r)$-approximate distance labeling scheme is referred to as an r-*additive* distance labeling scheme. An 0-additive (or 1-multiplicative) distance labeling is also termed an *exact* distance labeling scheme.

We consider two variations on the distance labeling definition: the *distinct* label model and the *non-distinct* label model. The latter one assumes that Eq. (1) holds only for $u \neq v$, that is we are not interested in the computation of $d_G(u,u)$. This allows to build schemes with sub-logarithmic label size. In the extreme case, graphs of constant diameter enjoy some constant additive distance labeling scheme with no labels at all in the non-distinct label model: it suffices to return a constant large enough to fulfill Eq. (1). On contrary, $\log n$ becomes a lower bound on the size of labels in the distinct label model, even in the case of n-node clique, since all the labels need to be different. We remark that any distance labeling scheme in the non-distinct label model can be transformed (within the same estimate quality) in the distinct label model adding an unique identifier of the nodes on $\log n$ bits. The notations $\ell_{\langle L,f \rangle}(\mathcal{G})$ and $\ell_{(s,r)}(\mathcal{G})$ are related to the non-distinct label model, and thus differs from the distinct label model by an additive factor $\log n$.

1.2 Related Works

Exact distance labeling scheme for weighted n-node trees with integral weights from the range $[0,W)$ using $O(\log^2 n + \log n \log W)$ bit labels has been given in [Pel99], and $O(\log^2 n)$ bit labels for interval graphs and permutation graphs were presented in [KKP00], all with $O(\log n)$ query time. The *Squashed Cube Conjecture* of Graham and Pollak [GP72], proved by Winkler [Win83], implies an exact distance labeling scheme for arbitrary n-node graphs with $n \log_2(3) \approx 1.58n$ bit labels for general graphs, although with a prohibitive $\Theta(n)$ query time to decode the distance. With a different approach, [GPPR00] presented a distance labeling scheme with label size $11n + o(n)$ and with $O(\log\log n)$ query time. In [Pel99], it is built a $8k$-multiplicative distance labeling scheme, for each integer $k \geqslant 1$, using $O(k\, n^{1/k} \log n \, \log W)$ bit labels with query time linear in the label size, where W stands for the integral weighted diameter. It follows for unweighted n-node graphs, a $O(\log n)$-multiplicative distance labeling scheme using $O(\log^3 n)$ bit labels with poly-log query time.

Some lower bounds on the size of the labels are established in [GPPR00] for several families of n-node graphs. In particular, it is shown that $\ell_{(s,0)}(\mathcal{G}_n) = \Omega(n)$ for every $s < 3$, where $\mathcal{G}_n$ denotes the family of all connected graphs. For the family $\mathcal{G}_{r(n),n}$ of graphs of treewidth bounded by $r(n)$, it is shown that $\ell_{(s,0)}(\mathcal{G}_{r(n),n}) = \Omega(r(n))$, for all $s < 3$ and $r(n) < n/2$. On the other side, it is proposed an exact distance labeling scheme for $\mathcal{G}_{r(n),n}$ with $O(R(n)\log n)$ bit labels and $O(r(n)\log n)$ query time, where $R(n) = \sum_{i=1}^{\log n} r(n/2^i)$. (We have $R(n) \leqslant r(n)\log n$, and for monotone $r(n) \geqslant n^\epsilon$ with constant $\epsilon > 0$, $R(n) = O(r(n))$.) It follows, for instance, that planar graphs support exact distance labeling with $O(\sqrt{n}\log n)$ bit labels, and that trees and bounded treewidth graphs support exact distance labeling with $O(\log^2 n)$ bit labels.

Finally, for the family $\mathcal{B}_n$ of bounded degree graphs, it is proved that $\ell_{(1,0)}(\mathcal{B}_n) = \Omega(\sqrt{n})$, and $\ell_{(1,0)}(\mathcal{P}_n) = \Omega(n^{1/3})$ for the family $\mathcal{P}_n$ of bounded degree planar graphs. (For the family $\mathcal{P}_{W,n}$ of bounded degree planar graphs with weights in $[0, \sqrt{n}]$, it is proved $\ell_{(1,0)}(\mathcal{P}_{W,n}) = \Omega(\sqrt{n})$.) Finally, for the family $\mathcal{T}_n$ of binary trees, we have developed a specific lower bound and we have showed $\ell_{(1,0)}(\mathcal{T}_n) \geqslant \frac{1}{8}\log^2 n - O(\log n)$. (For the family $\mathcal{T}_{W,n}$ of binary trees with weights from the range $[0, W)$, it is proved $\ell_{(1,0)}(\mathcal{T}_{W,n}) \geqslant \frac{1}{2}\log n \log W - O(\log W)$.) Actually, in [GPPR01] the *total* label length is lower bounded, the sum of the label size over all the nodes of the graph. It is shown that, excepted for trees, all the previous lower bounds hold simultaneously on $\Theta(n)$ of the graphs in the worst-case.

1.3 Our Contribution

Section 2 deals with bounds on s-multiplicative distance labeling schemes. It is first shown that planar graphs have a 3-multiplicative distance labeling scheme with $O(n^{1/3}\log n)$ bit labels. This should be contrasted with the fact that in [GPPR00] it is established that for exact distance labeling labels of size $\Omega(n^{1/3})$ are required, whereas $O(\sqrt{n}\log n)$ bit label suffice.

Then, we propose a $(1+1/\log W)$-multiplicative distance labeling scheme using $O(\log n \cdot \log\log W)$ bit labels is given for the family of weighted n-node trees with weighted diameter bounded by W. More generally, we show that the family of n-node graphs of treewidth at most $r(n)$ and weighted diameter bounded by W has a $(1+1/\log W)$-multiplicative distance labeling scheme using $O(R(n)\log\log W)$ bit labels and with $O(r(n))$ query time, assuming that $W = n^{O(1)}$. Hence, unweighted trees (and bounded treewidth graphs) enjoy a $(1+1/\log n)$-multiplicative distance labeling scheme using $O(\log n \cdot \log\log n)$ labels with constant query time.

We then turn to lower bounds on s-multiplicative distance labeling schemes. We establish, for every $s \geqslant 1$, a lower bound on such schemes on general n-node graphs. Specifically, it is shown that $\Omega(n^{1/(3s/4-O(1))})$ bit label for general graphs are required in the worst-case. The current upper bound is $O(s\, n^{1/\lceil s/8\rceil}\log^2 n)$, derived from the result of [Pel99] previously mentioned. Also, for trees, we show a lower bound of $\Omega(\log n \cdot \log\log n)$ bit label for every $(1+1/\log n)$-multiplicative scheme, proving by the way that our scheme for the upper bound is optimal. For the class of weighted trees, whose weights are in the range $[0, Z)$, for Z large enough, we show a lower bound of $\Omega(\log n \cdot \log\log Z)$ for $(1+1/\log Z)$-multiplicative scheme (still an optimal bound).

In Section 3.1, we turn to r-additive distance labeling schemes. We consider the family of k-chordal n-node graphs and diameter D, and show that they enjoy a $\lfloor k/2\rfloor$-additive distance labeling scheme with labels of size $O(\log n \log D)$. In particular, the family of n-node chordal graphs has an 1-additive distance labeling scheme with $O(\log^2 n)$ bit labels. Moreover, these schemes are polynomial-time constructible, and $O(1)$ query time. We show that the class of graphs of diameter D and having a dominating pair (class including AT-free, permutation, interval graphs) enjoys a 2-additive scheme, and even a 1-additive scheme for interval graphs, with $\log D + O(1)$ bits labels (in the non-distinct label model, and $O(\log n)$ bit label otherwise).

We then turn to lower bound for r-additive distance labeling schemes. We show, for every $r \geqslant 0$, a lower bound of $\Omega(\sqrt{n/(r+1)})$ for general n-node graphs (and $\Theta(n)$ if $r < 2$), and show a $\Omega(\log^2(n/(r+1)))$ lower bound for r-additive scheme on k-chordal, proving that our scheme for k-chordal graphs is optimal in size of the label and in the quality of the approximation. We also notice that AT-free graphs or k-chordal graphs required $\Omega(n)$ bit label for exact distance labeling scheme.

In Section 4, we conclude by some open problems.

2 Multiplicative Approximate Schemes

2.1 A Scheme for Planar Graphs

In this section we show that for the family of planar graphs enjoys a 3-multiplicative distance labeling schemes with $O(n^{1/3})$ bit labels. Note that exact distance labeling schemes require $\Omega(n^{1/3})$ bit labels, whereas $O(\sqrt{n}\log n)$ bit labels suffice.

Let us start with some background concerning partitions of planar graphs. A *region* is a connected subgraph of a planar graph. One can distinguish two kinds of nodes: internal (belonging to only one region) and boundary nodes (that belong to two or more regions).

The following decomposition lemma has been established in [FJ89] using the $O(\sqrt{n})$-separator theorem, and used in the analysis of compact routing in planar graphs.

Lemma 2.1 [FJ89] *For every n-node planar graph G and integer $k > 0$, it is possible (in polynomial time) to partition the nodes of G into k regions, each of $O(n/k)$ nodes and with $O(\sqrt{n/k})$ boundary nodes, such that any path connecting an internal node of one region to an internal node of another must go through at least one boundary node of each region.*

Setting $k = \lceil n^{1/3} \rceil$, we get $O(n^{1/3})$ regions $R_1, \dots, R_k$, each with $O(n^{2/3})$ nodes and $O(n^{1/3})$ boundary nodes. Now, each region R_i is partitioned again into $O(n^{1/3})$ subregions S_j of $O(n^{1/3})$ (internal and boundary) nodes.

Each node is given a unique identifier, i.e., an integer $I(v) \in \{1, \dots, n\}$, as well as a pair $r(v) = (i, j)$, indicating its region number i and its subregion number j. (For boundary nodes, choose a valid pair (i, j) arbitrarily.)

Consider a node u that belongs to a region R_i and to the subregion S_j within R_i. For every other region $R_{i'}$ of G, $i' \neq i$, let $p_u(R_{i'})$ denote the closest node to u in $R_{i'}$. Note that $p_u(R_{i'})$ is necessarily a boundary node in $R_{i'}$. Similarly, for every other subregion $S_{j'}$ of R_i, $j' \neq j$, let $p_u(S_{j'})$ denote the closest node to u in $S_{j'}$.

The label $L(u, G)$ assigned to the node u consists of the following fields:

[a] its identifier $I(u)$ and pair $r(u) = (i, j)$;
[b] the distance from u to all the nodes in S_j;
[c] the identifier of $p_u(R_{i'})$ and the distance $d_G(u, p_u(R_{i'}))$, for every region $R_{i'}$ of G, $i' \neq i$;
[d] the identifier of $p_u(S_{j'})$ and the distance $d_G(u, p_u(S_{j'}))$, for every subregion $S_{j'}$ of R_i, $j' \neq j$;
[e] the distance from u to all the boundary nodes in its region R_i.

The number of bits in the resulting label is bounded by $O(n^{1/3} \log n)$. The distance between u and v is computed from the labels $L(u, G)$ and $L(v, G)$ as follows:

1. Extract the region and subregion numbers of u and v from field [a] of the two labels.
2. If u and v belong to the same subregion S_j in the same region R_i, then do:
   (a) extract $d_G(u, v)$ from field [b] of $L(v, G)$ using $I(v)$;
   (b) set $\tilde{d}(u, v) = d_G(u, v)$.
3. If u and v are in the same region R_i but in different subregions, say $u \in S_j$ and $v \in S_{j'}$, then do:
   (a) extract $z = p_u(S_{j'})$ and $d_G(u, z)$ from field [d] of $L(u, G)$;
   (b) extract $d_G(z, v)$ from field [b] of $L(v, G)$;
   (c) set $\tilde{d}(u, v) = d_G(u, z) + d_G(z, v)$.
4. If u and v belong to different regions, say $u \in R_i$ and $v \in R_{i'}$, then do:
   (a) extract $z = p_u(R_{i'})$ and $d_G(u, z)$ from field [c] of $L(u, G)$;
   (b) extract $d_G(z, v)$ from field [e] of $L(v, G)$;
   (c) set $\tilde{d}(u, v) = d_G(u, z) + d_G(z, v)$.
5. Return $\tilde{d}(u, v)$.

Theorem 2.2 *There exists a 3-multiplicative distance labeling scheme using labels of size $O(n^{1/3} \log n)$ for the family of planar graphs with n nodes. Moreover the labels are polynomial time constructible and the distance decoder has time complexity $O(1)$.*

Proof. Clearly, in case u and v belong to the same subregion S_j of the same region R_i, $\tilde{d}(u, v)$ returns the exact distance. By the triangle inequality, and the separator property of internal nodes, it is easy to show that in case u and v belong to the same subregion S_j in the same region R_i,

$$d_G(u, v) \leqslant d_G(u, p_u(S_{j'})) + d_G(p_u(S_{j'}), v) \leqslant 3 \cdot d_G(u, v) .$$

A similar argument applies for the case where u and v belong to different regions. Since all the tables used contain at most $O(n^{1/3})$ elements of $O(\log n)$ bits, we can implement them in order to guarantee extraction in $O(1)$ time using perfect hashing functions (cf. [FKS84]). Note that the region decomposition and hashing tables are polynomial time constructible, thus the labels too. ∎

2.2 A Scheme for Trees and Bounded Treewidth Graphs

A pair of integer functions $\langle \lambda, \phi \rangle$ is an (s,r)-*estimator* of $\{1,\dots,W\}$ if $\lambda : \{1,\dots,W\} \to \{1,\dots,2^\alpha\}$ (where typically $2^\alpha \ll W$), $\phi : \{1,\dots,2^\alpha\} \to \mathbb{N}$, and for every $x \in \{1,\dots,W\}$, $x \leqslant \phi(\lambda(x)) \leqslant s \cdot x + r$. Intuitively, we think of λ as a function "compacting" x, and of ϕ as a function attempting to reconstruct x from $\lambda(x)$. The *size* of the estimator is α, and its *time complexity* is the worst-case time complexity of the function ϕ.

Lemma 2.3 *For every $k \leqslant O(\omega)$, and for every $m \in \{0,\dots,k\}$, there exists a constant time $(1+2^{-m},0)$-estimator of $\left\{1,\dots,2^k\right\}$ of size $\alpha = m + \lceil \log(k-m+1) \rceil$.*

Proof. Consider some $x \in \left\{1,\dots,2^k\right\}$, and let $S = S_k \cdots S_1$ be the binary representation of $x-1$. Let j be the index of the most significant bit of S set to 1, i.e., $S_k = \dots = S_{j+1} = 0$ and $S_j = 1$. (For $x = 1$ we set $j = 1$). Let $p = \max\{j-m, 0\}$, $p \in \{0,\dots,k-m\}$. Let t denote the $(j-p)$-bit number represented by the substring $S_j \cdots S_{p+1}$. Note that $t \in \{0,\dots,2^m-1\}$ since $p \geqslant j-m$.

Set $\lambda(x) = \langle t, p \rangle$. Clearly, $\lambda(x)$ is of size at most $m + \lceil \log(k-m+1) \rceil = \alpha$.

Setting $\phi(\langle t,p \rangle) = (t+1) \cdot 2^p$ (computed in constant time for $k \leqslant O(\omega)$), we have that $x \leqslant \phi(\lambda(x)) \leqslant x + 2^p - 1 < x(1 + 2^p/x)$. Because $x \leqslant (t+1) \cdot 2^p$ and $t \leqslant 2^m - 1$, we conclude that $1 + 2^p/x \leqslant 1 + 2^{-m}$, and thus that $\langle \lambda, \phi \rangle$ is a $(1+2^{-m},0)$-estimator. ∎

Given a binary string S of length k, let $\mathrm{rank}_S(i)$ denote the function that returns the number of 1's up to and including position i, for $i \in \{1,\dots,k\}$, and let $\mathrm{lsb}_k(S)$ denote the position p of the least significant bit in S set to 1, $p \in \{0,\dots,k\}$ ($p = 0$ if S is null). The two following results require constant time integer multiplications and divisions on $O(\log n)$-bit words, which are allowed in our computational model.

Lemma 2.4

[**Mun96, MR97**] *For every integer $k \leqslant n$, and for every binary string S of length k, the operation* $\mathrm{rank}_S(\cdot)$ *can be performed in $O(1)$ worst-case time with $o(k)$ pre-computed auxiliary bits, and $k + o(k)$ bits in total.*

[**FW93**] *For every integer $k \leqslant O(\log n)$, the operation* $\mathrm{lsb}_k(\cdot)$ *can be performed in $O(1)$ worst-case time with $O(k)$ pre-computed auxiliary bits.*

We note that the results of [Mun96, MR97] require essentially bucketing with shifting on binary words, and that $\mathrm{lsb}_k(\cdot)$ can be computed with shifting and boolean bitwise operations only in $O(\log\log k)$ time.

We state the main result for weighted trees. Given an integer $W > 0$, a W-*tree* is a weighted tree whose weighted diameter is bounded by W. Note that an unweighted tree T of diameter D is a particular case of a W-tree, for $W \geqslant D$.

Theorem 2.5 *Let α be the size of an (s,r)-estimator of $\{1,\ldots,W\}$, and let t denote its time complexity. Then there exists an $(s,2r)$-approximate distance labeling scheme using labels of size $\alpha\log n + O(\log n)$ for the family of W-trees with n nodes. Moreover the distance decoder has time complexity $O(t)$.*

Proof. It is well-known that every n-node tree has a node, hereafter called a *separator*, which splits tree into connected components each of at most $n/2$ nodes.

Let T be any n-node W-tree, and $u \in V(T)$, and let $\langle\lambda,\phi\rangle$ be an (s,r)-estimator of $\{1,\ldots,W\}$. Apply the above separator property to T, and let s_1 be the obtained separator. With each connected component F of $T\setminus\{s_1\}$, we associate a unique label $c(F)\in\{1,\ldots,\deg(s_1)\}$ such that for every two components A,B of $T\setminus\{s_1\}$, if $|V(A)|\geqslant|V(B)|$ then $c(A)\leqslant c(B)$. Let T_1 be the connected component of $T\setminus\{s_1\}$ containing u, and let $c_1 = c(T_1)$.

We recursively apply this decomposition scheme to T_i, $i\geqslant 1$, in order to obtain s_{i+1}, T_{i+1} and c_{i+1} such that $u\in V(T_{i+1})$ and $c_{i+1} = c(T_{i+1})$, until we have $u = s_{i+1}$. Let h be the index such that $u = s_{h+1}$. Note that $h+1\leqslant\log n$. We set the label of u in T to be

$$L(u,T) \;=\; \langle (c_1,\ldots,c_h),(\lambda(d_T(u,s_1)),\ldots,\lambda(d_T(u,s_h)))\rangle\,.$$

Given two nodes u,u' of labels $\langle(c_1,\ldots,c_h),(\lambda_1,\ldots,\lambda_h)\rangle$ and $\langle(c'_1,\ldots,c'_{h'}),(\lambda'_1,\ldots,\lambda'_{h'})\rangle$ respectively, we compute the distance as follows:

1. compute the lowest index i_0 such that $c_{i_0}\neq c'_{i_0}$;
2. return $\phi(\lambda_{i_0})+\phi(\lambda'_{i_0})$.

To prove the correctness of the scheme, it suffices to observe that s_{i_0} is the lowest common ancestor of u and u' in T. The scheme is therefore $(s,2r)$-approximate.

Consider the node u. Since the size of the estimator $\langle\lambda,\phi\rangle$ is α, the sequence $\Lambda = (\lambda_1,\ldots,\lambda_h)$ can be stored for u with αh bits.

Let us now give a compact representation for the sequence $C=(c_1,\ldots,c_h)$ of u. Let $l_i = \max\{\lceil\log c_i\rceil,1\}$ denote the length of the integer c_i-1 written in binary (recall that $c_i\geqslant 1$). The sequence C can be coded with two binary strings, say A and B, each of length $\sum_{i=1}^h l_i$. A is composed of the concatenation of the c_i's written on l_i bits, and B is a bitmap indicating the beginning of each field.

Note that $c_i\geqslant 1$ implies $1\leqslant l_i\leqslant 1+\log c_i$. Hence, $\sum_{i=1}^h l_i\leqslant\log\left(\prod_{i=1}^h c_i\right)+h$. For every $i\geqslant 0$, let $n_i = |V(T_i)|$, setting $n_0 = |V(T)| = n$. Due to the rule assigning the labels c_i we have, for every $i>0$, $n_i\leqslant n_{i-1}/c_i$ (otherwise T_{i-1} would have more than n_{i-1} nodes). Therefore, $c_i\leqslant n_{i-1}/n_i$ and hence $\prod_{i=1}^h c_i\leqslant n_0 = n$. It follows that C can be coded with at most $2\sum_{i=1}^h l_i\leqslant 2\log n+2h$ bits. In total, C and Λ can be coded with $\alpha h+2\log n+2h\leqslant\alpha\log n+4\log n$ bits. To perform a fast computation of the distance, we store $O(\log n)$ extra bits using the method of Lemma 2.4. Therefore the length of the label of u is bounded by $\alpha\log n+O(\log n)$.

The complexity of the distance decoder is bounded by $2t$ plus the time to compute the lowest integer i_0, given the binary strings (A, B) for the node u, and (A', B') for u'. For our purpose, we assume that the binary strings A, B, A', B' are stored in reverse order, i.e., the most significant bit becomes the rightmost bit. Moreover, we assume that these strings are all of exactly $k = \lfloor 2 \log n \rfloor$ bits, padding them by some extra 0's to the left as needed (recall that originally each of the strings is of length $\sum_{i=1}^{h} l_i \leqslant 2 \log n$).

First, compute h and h' noting that $h = \mathrm{rank}_B(k)$ and $h' = \mathrm{rank}_{B'}(k)$. Now, assume that $h \geqslant h'$ (the computation of i_0 is similar if $h < h'$). Then, compute

$$a = \mathrm{rank}_B(\mathrm{lsb}_k(A \operatorname{xor} A')) \quad \text{and} \quad b = \mathrm{rank}_B(\mathrm{lsb}_k(B \operatorname{xor} B')),$$

where xor denotes the bitwise exclusive-or operation on words of length k. It is easy to see that if $B \neq B'$ then $h - b + 1$ is the lowest index of field where the sequences B and B' differ, and if $B = B'$, i.e., the fields are all of equal length, then $h - a + 1$ is the lowest index of field where A and A' differ. Therefore, if $B = B'$ then $i_0 = h - a + 1$, and otherwise $i_0 = \min\{h - a, h - b\} + 1$. By Lemma 2.4, the time to compute i_0 is bounded by a constant, and the number of additional bits required to perform fast queries for $\mathrm{rank}_B(\cdot)$ and $\mathrm{lsb}_k(\cdot)$ is at most $k + o(k) + O(k) = O(\log n)$. This completes the proof. ∎

In particular, choosing $k = \lceil \log W \rceil$ and $m = \lceil \log \log W \rceil$ in Lemma 2.3, we have: (note that if $W = n^{O(1)}$, then $k = O(\omega)$ and thus all the distances can be estimated in constant time)

Corollary 2.6 *There exists a $(1 + 1/\log W)$-multiplicative distance labeling scheme using labels of size $O(\log n \cdot \log \log W)$ for the family of W-trees with n nodes. Moreover the distance decoder has time complexity $O(1)$ for $W = n^{O(1)}$.*

For unweighted trees we have:

Corollary 2.7 *There exists a $(1 + 1/\log n)$-multiplicative distance labeling scheme using labels of size $O(\log n \cdot \log \log n)$ for the family of trees with n nodes. Moreover the distance decoder has time complexity $O(1)$.*

The scheme for trees can be apply to tree-decomposition used in the frame of bounded treewidth graphs [RS86] (see also Section 3.1). In [GPPR00], it is shown that graphs of treewidth bounded by $r(n)$ have exact distance labeling scheme using $O(R(n) \log n)$ bit labels, for every monotone function $r(n)$, and with $O(r(n) \log n)$ query time, where $R(n) = \sum_{i=0}^{\log n} r(n/2^i)$. Using the above scheme for trees, we can extend this result.

Corollary 2.8 *There exists a $(1 + 1/\log W)$-multiplicative (resp. exact) distance labeling scheme using labels of size $O(R(n) \log \log W)$ (resp. $O(R(n) \log W)$) for the family of weighted n-node graphs of treewidth at most $r(n)$ and of weighted diameter W. Moreover the distance decoder has time complexity $O(r(n))$ for $W = n^{O(1)}$.*

Therefore n-node trees (and bounded treewidth graphs) enjoy exact distance labeling with $O(\log^2 n)$ bit labels with constant query time (improving query time of the scheme of [Pel99]).

2.3 A Lower Bound for Multiplicative Schemes

Instead of considering the maximal label length one can prefer the *total* label length. For every graph family $\mathcal{F}$ under consideration, we assume that each n-node graph of $\mathcal{F}$ is a labeled graph on the set of nodes $V_n = \{1,\ldots,n\}$. For every node subset $A \subseteq V_n$, denote

$$\overline{\ell}_{\langle L,f\rangle}(A,\mathcal{F}) \;=\; \max_{G\in\mathcal{F}} \sum_{u\in A} |L(u,G)|$$

and

$$\overline{\ell}_{(s,r)}(A,\mathcal{F}) \;=\; \min_{\langle L,f\rangle} \left\{\overline{\ell}_{\langle L,f\rangle}(A,\mathcal{F})\right\}$$

where the minimum is taken over all (s,r)-approximate distance labeling schemes $\langle L,f\rangle$ for $\mathcal{F}$. For simplicity, let $\overline{\ell}_{(s,r)}(\mathcal{F}) = \overline{\ell}_{(s,r)}(V_n,\mathcal{F})$.

To prove a lower bound on s-multiplicative distance labeling schemes on general graphs, we need the following concept introduced in [GPPR00]. Let $A \subseteq V_n$, and let $k > 1$ be a real number (k can be a function of n). Consider a family $\mathcal{F}$ of labeled graphs on the set of nodes V_n. The graph family $\mathcal{F}$ is an (A,k)-*family* if for every two distinct graphs $G, H \in \mathcal{F}$ there exist $x, y \in A$ such that $d_G(x,y) \geqslant k \cdot d_H(x,y)$ or $d_H(x,y) \geqslant k \cdot d_G(x,y)$. The following is shown in [GPPR00].

Lemma 2.9 [GPPR00] *Let $\mathcal{F}$ be an (A,k)-family for $k > 1$. Then for any $s < k$,*

$$\ell_{(s,0)}(\mathcal{F}) \;\geqslant\; \frac{1}{|A|}\cdot\log|\mathcal{F}| \quad \textit{and} \quad \overline{\ell}_{(s,0)}(A,\mathcal{F}) \;\geqslant\; \log|\mathcal{F}| - |A|\log\log|\mathcal{F}| \;.$$

We need also of the following result:

Lemma 2.10 *Let G be any connected graph with n nodes, m edges, and girth g. Then the family $\mathcal{S}_G$ composed of all the n-node connected subgraphs of G satisfies, for every real number $1 \leqslant s < g-1$,*

$$\ell_{(s,0)}(\mathcal{S}_G) \;\geqslant\; \frac{m+1}{n} - 1 \quad \textit{and} \quad \overline{\ell}_{(s,0)}(\mathcal{S}_G) \;\geqslant\; m - 2n\log n \;.$$

Proof. Let $V_n = \{1,\ldots,n\}$ be the set of nodes of G, and let $\mathcal{S} = \mathcal{S}_G$. Let us show that $\mathcal{S}$ is a $(V_n, g-1)$-family. Let H_1, H_2 be two distinct (connected) graphs of $\mathcal{S}$. Necessarily, H_1 and H_2 differ by at least one edge. Without loss of generality assume that $\{x,y\} \in E(H_1)$ and $\{x,y\} \notin E(H_2)$. Because $\{x,y\}$ belongs in G to a chordless cycle of length at least

g, it follows that $d_{H_1}(x,y) = 1$ whereas $d_{H_2}(x,y) \geqslant g-1 = (g-1)\cdot d_{H_1}(x,y)$. It turns out that $\mathcal{S}$ is a $(V_n, g-1)$-family of connected graphs. By Lemma 2.9 we have for every $s < g-1$,

$$\ell_{(s,0)}(\mathcal{S}) \;\geqslant\; \frac{1}{n}\log|\mathcal{S}| \quad \text{and} \quad \overline{\ell}_{(s,0)}(\mathcal{S}) \;\geqslant\; \log|\mathcal{S}| - n\log\log|\mathcal{S}| \;.$$

Clearly, $|\mathcal{S}| \geqslant 2^{m-(n-1)}$ (to see this, it suffices to consider any spanning tree T of G; $\mathcal{S}$ contains at least all the possible subgraphs of G that contain the edges of T). We complete the proof noting that $|\mathcal{S}| \leqslant 2^{n^2/2}$ and that $m-(n-1)-n\log(n^2/2) \geqslant m - n\log(n^2)$. ∎

In particular, Lemma 2.10 implies that, for every $s < g-1$, $\ell_{(s,0)}(\mathcal{G}_n) \geqslant m(g,n)/n - O(1)$, where $m(g,n)$ denotes the maximum number of edges in an n-node graphs of girth g. In [ERS66] it is conjectured that $m(2k+2,n) = \Omega(n^{1+1/k})$, proving later for $k = 1,2,3$ and 5. In [LUW95], it is shown that $m(4k+2,n) = \Omega(n^{1+1/(3k-1)})$ and that $m(4k,n) = \Omega(n^{1+1/(3(k-1))})$. It follows,

Theorem 2.11 *For every $s \geqslant 1$,*

$$\ell_{(s,0)}(\mathcal{G}_n) \;=\; \Omega\left(n^{1/(3s/4-O(1))}\right) \quad \textit{and} \quad \overline{\ell}_{(s,0)}(\mathcal{G}_n) \;=\; \Omega\left(n^{1+1/(3s/4-O(1))}\right) \;.$$

2.4 A Lower Bound on Trees

In this section we prove two lower bounds on trees. First, we show that for the family $\mathcal{T}_{Z,n}$ of weighted binary n-node trees with weights from the range $[0,Z)$ for sufficiently large Z, any s-multiplicative distance labeling scheme, for constant s, requires $\ell_{(s,0)}(\mathcal{T}_{Z,n}) = \Omega(\log n(\log\log Z + \log\log n))$. Then, we show that for the family $\mathcal{T}_n$ of unweighted binary n-node trees, any $(1+1/\log n)$-multiplicative distance labeling scheme requires $\ell_{(1+1/\log n,0)}(\mathcal{T}_n) = \Omega(\log n \cdot \log\log n)$. Both bounds are tight given the distance labeling schemes presented earlier (cf. corollaries 2.6 and 2.7).

2.4.1 The class of trees

Let us first define the sets of weights used for the proof. We start by setting, for any integer $M \geqslant 2$, and for any real $\epsilon > 0$, the values $\gamma_0(M) = 0$, $\beta = 1+2\epsilon$ and $A = (\beta/\epsilon)^2$. We next define, for every $\mu \geqslant 1$,

$$\begin{aligned}
Q_\mu(M) &= \left\{ \left\lceil A\cdot\beta^{2(\mu-1)M+j}\right\rceil \mid 0 \leqslant j \leqslant M-1 \right\}, \\
\gamma_\mu(M) &= \max Q_\mu(M) \;=\; \left\lceil A\cdot\beta^{2\mu M-1}\right\rceil, \\
Z_\mu(M) &= \gamma_\mu(M) - \gamma_{\mu-1}(M).
\end{aligned}$$

For the lower bound proof we focus on a special family of binary weighted trees called (μ,M)-*trees*, defined as follows. For $\mu = 1$, a $(1,M)$-tree T is composed of a root with a single child and two grandchildren. An integral weight $x \in Q_1(M)$ is associated with each of the two edges connecting the child to the two grandchildren, and the weight $Z_1(M) - x$ is associated with the edge connecting the root to the child. (See Fig. 1.)

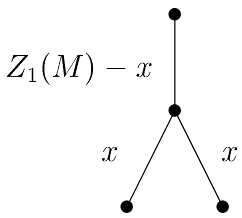

Figure 1: A $(1, M)$-tree.

For $\mu \geqslant 2$, a (μ, M)-tree is constructed as follows. Start by constructing a $(1, M)$-tree, except that the weight assigned to the two edges leading to its leaves is taken to be some $x \in Q_\mu(M)$, and the weight of the edge emanating the root is $Z_\mu(M) - x$. Then, attach a $(\mu - 1, M)$-tree to each of the two leaves. Hence a (μ, M)-tree contains 2^μ leaves, denoted $a_1, \ldots, a_{2^\mu}$. Let $\mathcal{C}(\mu, M)$ denote the family of all (μ, M)-trees. Note that all of those trees have the same structure, and they differ only in their weight assignment. Fig. 2 depicts a $(3, M)$-tree.

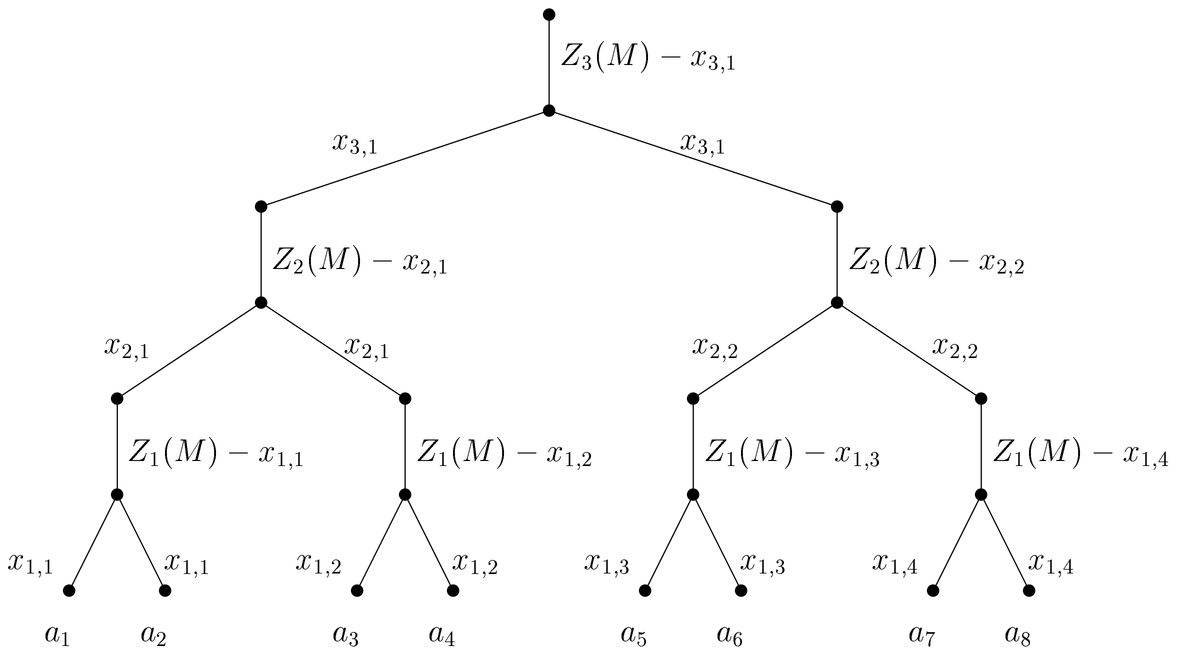

Figure 2: A $(3, M)$-tree.

Note that a (μ, M)-tree T is completely defined by the triple $T = (T_0, T_1, x)$, where x is the weight associated with the two edges of the top $(1, M)$-tree, and T_0 and T_1 are the two $(\mu - 1, M)$-trees attached to the leaves of the top tree. The subfamily of $\mathcal{C}(\mu, M)$

consisting of (μ, M)-trees with topmost weight x is denoted $\mathcal{C}(\mu, M, x)$. Hence

$$\mathcal{C}(\mu, M) \;=\; \bigcup_{x \in Q_\mu(M)} \mathcal{C}(\mu, M, x) \;.$$

By the definition of these binary trees we have

Lemma 2.12 *For every two leaves a, a' of a tree $T \in \mathcal{C}(\mu, M, x)$,*

- *If $a \in T_0$ and $a' \in T_1$ then $d_T(a, a') = 2\gamma_{\mu-1} + 2x$,*
- *If $a, a' \in T_i$ (for $i \in \{0,1\}$) then $d_T(a, a') = d_{T_i}(a, a')$.* ∎

This implies the following lemma.

Lemma 2.13 *Consider two (μ, M)-trees $T = (T_0, T_1, x)$ and $T' = (T'_0, T'_1, x')$. For any leaves $a_0 \in T_0$, $a_1 \in T_1$, $a'_0 \in T'_0$ and $a'_1 \in T'_1$,*

$$d_T(a_0, a_1) = d_{T'}(a'_0, a'_1) \quad \Longleftrightarrow \quad x = x' \;. \quad ∎$$

Let T be a (μ, M)-tree, and let a and a' be two leaves of T. We say that a and a' are *x-separated on level t* of T if they both belong to the same (t, M)-subtree T', but to different $(t-1, M)$-subtrees T'_0 and T'_1, and moreover, the weight of the two edges connecting the roots of T'_0 and T'_1 to their parent in T' is x. The two leaves of a $(1, M)$-tree in $\mathcal{C}(1, M, x)$ are x-separated on level 1.

Hence another implication of Lemma 2.12 is the following.

Corollary 2.14 *For every two leaves a, a' of a (μ, M)-tree T, if a and a' are x-separated on level t, then $d_T(a, a') = 2\gamma_{t-1} + 2x$.* ∎

2.4.2 The proof

A $(1+\epsilon, 0)$-*approximate* (μ, M)-*legal* distance labeling scheme is a scheme $\langle L, f \rangle$ that correctly provides a $(1+\epsilon, 0)$-approximation for the distance between leaves on all (μ, M)-trees. Namely, L is a node-labeling assigning a label $L(a, T)$ to each leaf a of any (μ, M)-tree T, such that for every two leaves a and a' with labels $\lambda = L(a, T)$ and $\lambda' = L(a', T)$, f computes a $(1+\epsilon, 0)$-approximate estimate $f(\lambda, \lambda') = \tilde{d}(a, a')$ for the distance, satisfying

$$d_T(a, a') \;\leqslant\; \tilde{d}(a, a') \;\leqslant\; (1+\epsilon) \cdot d_T(a, a') \;.$$

Note that by Corollary 2.14, this $(1+\epsilon, 0)$-approximate estimate must be taken from a specific range of possible values. In particular, if a and a' are x-separated on level t, then the allowed range for the distance estimate is

$$R(t, x) = [2\gamma_{t-1} + 2x \;,\; (1+\epsilon)(2\gamma_{t-1} + 2x)] \;.$$

Lemma 2.15 *For every $x, x' \in Q_t(M)$, if $x' \neq x$ then $R(t,x) \cap R(t,x') = \varnothing$.*

Proof. Consider two different weights $x, x' \in Q_t(M)$ where without loss of generality $x = \lceil A \cdot \beta^{2(\mu-1)M+j} \rceil$ and $x' = \lceil A \cdot \beta^{2(\mu-1)M+j'} \rceil$ for $j < j'$. Then the lowest possible $(1+\epsilon, 0)$-approximate estimate for x' in $R(t,x')$ is

$$\tilde{d}_{x'} \;=\; 2\gamma_{t-1} + 2x' \;\geqslant\; 2A\left(\beta^{2(t-1)M-1} + \beta^{2(t-1)M+j'}\right),$$

and the highest possible $(1+\epsilon, 0)$-approximate estimate for x in $R(t,x)$ is

$$\tilde{d}_x \;=\; (1+\epsilon)(2\gamma_{t-1} + 2x) \;<\; 2(1+\epsilon)\left(A \cdot \beta^{2(t-1)M-1} + A \cdot \beta^{2(t-1)M+j} + 2\right).$$

It remains to verify that $\tilde{d}_x < \tilde{d}_{x'}$. After simplifying, it remains to show that

$$\epsilon A \cdot \beta^{2(t-1)M-1} + (1+\epsilon)A \cdot \beta^{2(t-1)M+j} + 2(1+\epsilon) \;<\; A \cdot \beta^{2(t-1)M+j'}.$$

As $j' \geqslant j+1$, it suffices to show that

$$A\epsilon \cdot \beta^{2(t-1)M-1} + 2(1+\epsilon) \;<\; A\,(\beta - 1 - \epsilon) \cdot \beta^{2(t-1)M+j},$$

or, that

$$A\epsilon\left(\beta^{2(t-1)M+j} - \beta^{2(t-1)M-1}\right) \;>\; 2(1+\epsilon).$$

As $A\epsilon > 1 + 1/\epsilon = 2(1+\epsilon) \cdot \frac{1}{2\epsilon}$, the above readily follows from the fact that

$$\beta^{2(t-1)M+j} - \beta^{2(t-1)M-1} \;\geqslant\; (\beta - 1) \cdot \beta^{2(t-1)M-1} \;\geqslant\; 2\epsilon\,. \qquad \blacksquare$$

Lemma 2.16 *Consider a (μ, M)-tree T, and let a and a' be two leaves x-separated on level t of T. Suppose that $\tilde{d}$ is a $(1+\epsilon, 0)$-approximate estimate for their distance $d_T(a,a') = 2\gamma_{t-1} + 2x$. Then given $\tilde{d}$ (and knowing M and ϵ) it is possible to extract the values of t and x, hence also the exact distance.*

Proof. Observe that the smallest estimate possible for any two leaves separated on level t of T, for any x, is

$$\tilde{d}_{min}(t) \;=\; 2\left(\gamma_{t-1} + \lceil A \cdot \beta^{2(t-1)M} \rceil\right) \;\geqslant\; 2A\left(\beta^{2(t-1)M-1} + \beta^{2(t-1)M}\right),$$

and the largest estimate possible is

$$\tilde{d}_{max}(t) \;=\; 2(1+\epsilon)\left(\gamma_{t-1} + \lceil A \cdot \beta^{(2t-1)M-1} \rceil\right) \;\leqslant\; 2(1+\epsilon)\left(A \cdot \beta^{2(t-1)M-1} + A \cdot \beta^{(2t-1)M-1} + 2\right).$$

We now claim that $\tilde{d}_{max}(t-1) < \tilde{d}_{min}(t)$. To prove this, it suffices to show that

$$A\left(\beta^{2(t-1)M-1} + \beta^{2(t-1)M}\right) \;>\; (1+\epsilon)\left(A \cdot \beta^{2(t-2)M-1} + A \cdot \beta^{(2t-3)M-1} + 2\right)\;.$$

Setting $\eta = \beta^{(2t-2)M}$, we have to show that

$$A(1+\beta)\eta \ > \ (1+\epsilon)\left(A\eta + A\eta/\beta^M + 2\beta\right),$$

or after simplifying, that

$$A\eta\left(1 - 1/\beta^M\right) \ > \ 2\beta.$$

Indeed, as $\eta \geqslant 1$ and $\epsilon \leqslant 1$,

$$A\eta\left(1-\beta^M\right) \ > \ \left(\frac{\beta}{\epsilon}\right)^2\left(1-\frac{1}{\beta}\right) \ = \ \frac{2\beta}{\epsilon} \ \geqslant \ 2\beta.$$

This implies that there is a complete separation between levels, hence t can be deduced with certainty given $\tilde{d}$. Once t is known, x can be inferred as well, by employing Lemma 2.15. ∎

Corollary 2.17 *If $\langle L, f\rangle$ is a $(1+\epsilon, 0)$-approximate (μ, M)-legal distance labeling scheme, then it provides also a $(1,0)$-approximate (μ, M)-legal distance labeling scheme.*

For a $(1+\epsilon, 0)$-approximate (μ, M)-legal labeling scheme, let $W(L,\mu,M)$ denote the set of all labels assigned by L to nodes in trees of $\mathcal{C}(\mu, M)$, and let $g(\mu, M)$ denote the minimum cardinality $|W(L,\mu,M)|$ over all $(1+\epsilon,0)$-approximate (μ,M)-legal labeling functions $\langle L, f\rangle$.

Hereafter, we fix $\langle L, f\rangle$ to be some $(1+\epsilon,0)$-approximate (μ, M)-legal distance labeling scheme attaining $g(\mu, M)$, i.e., such that $|W(L,\mu,M)| = g(\mu,M)$.

Let $W(\mu, M, x)$ denote the set of possible pairs of labels (λ_0, λ_1) assigned by L to some leaves $a \in T_0$ and $a' \in T_1$ respectively, for some tree $T = (T_0, T_1, x) \in \mathcal{C}(\mu, M, x)$. Let $\mathcal{W} = \bigcup_{x \in Q_\mu(M)} W(\mu, M, x)$. As $\mathcal{W} \subseteq W(L,\mu,M) \times W(L,\mu,M)$ we have

Lemma 2.18 $|\mathcal{W}| \leqslant g(\mu, M)^2$.

Lemma 2.19 *For every $x, x' \in Q_\mu(M)$, if $x' \neq x$ then $W(\mu,M,x) \cap W(\mu,M,x') = \varnothing$.*

Proof. Consider two different weights $x, x' \in Q_\mu(M)$. Assume by way of contradiction that there exists a pair $(\lambda_1, \lambda_2) \in W(\mu,M,x) \cap W(\mu,M,x')$. Then there exist two $(\mu-1, M)$-trees T_0, T_1 such that $T = (T_0, T_1, x)$ uses the label λ_1 for some leaf $a_{j_1} \in T_0$ and the label λ_2 for some leaf $a_{j_2} \in T_1$, and there exist two $(\mu-1, M)$-trees T_0', T_1' such that $T' = (T_0', T_1', x')$ uses the label λ_1 for some leaf $a_{j_3} \in T_0'$ and the label λ_2 for some leaf $a_{j_4} \in T_1'$. Therefore, querying f for the approximate distance between a_{j_1} and a_{j_2} will yield $f(\lambda_1, \lambda_2)$, and the same response will result from querying the distance between a_{j_3} and a_{j_4}.

But the range of possible $(1+\epsilon, 0)$-approximate distance estimates permissible for a_{j_1} and a_{j_2} is the set $R(\mu, x)$, and the range for a_{j_3} and a_{j_4} is the set $R(\mu, x')$, and these sets are disjoint by Lemma 2.15, contradiction. ∎

Lemma 2.20 *For every $x \in Q_\mu(M)$, $|W(\mu, M, x)| \geqslant g(\mu - 1, M^2)$.*

Proof. We derive a legal labeling scheme for all $(\mu-1, M^2)$-trees using at most $|W(\mu, M, x)|$ labels.

Consider a $(\mu - 1, M^2)$-tree T. Make two copies of the tree T, denoted T_0 and T_1, with weights modified as follows. Consider an edge e of T on level t, with weight $w \in Q_t(M^2)$. This weight is of the form $w = \left\lceil A \cdot \beta^{2(\mu-1)M^2+k} \right\rceil$ for some $0 \leqslant k \leqslant M^2 - 1$. Write k as $k = k_0 + Mk_1$ where $k_0 = k \bmod M$ and $k_1 = \lfloor k/M \rfloor$ (with $0 \leqslant k_0, k_1 \leqslant M-1$). Then assign the corresponding edge $e_0 \in T_0$ (respectively, $e_1 \in T_1$) the weight $w_0 = \lceil A \cdot \beta^{2(\mu-1)M+k_0} \rceil$ (resp., $w_1 = \lceil A \cdot \beta^{2(\mu-1)M+k_1} \rceil$).

Notice that T_0 and T_1 are $(\mu - 1, M)$-trees, and therefore can be combined into a (μ, M)-tree $\hat{T}$ by adding an appropriate $(1, M)$-tree joining their roots. Setting the weight of the connecting edges to x, the resulting (μ, M)-tree $\hat{T} = (T_0, T_1, x)$ belongs to the class $\mathcal{C}(\mu, M, x)$.

Number the leaves of T from left to right as $a_1, \ldots, a_{2^{\mu-1}}$. Similarly, number the leaves of T_0 (resp., T_1) from left to right as $a_1^0, \ldots, a_{2^{\mu-1}}^0$ (resp., $a_1^1, \ldots, a_{2^{\mu-1}}^1$). Every leaf a_j of T is now associated with two homologous leaves of $\hat{T}$, namely, the corresponding leaf a_j^0 occurring in T_0, and the leaf a_j^1 occurring in T_1.

Apply the $(1 + \epsilon)$-approximate (μ, M)-legal labeling L to label the tree $\hat{T} = (T_0, T_1, x)$. Now use the resulting labeling to define a labeling function L' for the nodes of T as follows. A leaf $a_j \in T$ receives as its label the pair $L'(a_j, T) = \left\langle L(a_j^0, T), L(a_j^1, T) \right\rangle$. Note that this pair belongs to $W(\mu, M, x)$.

The distance decoder f' for $(\mu - 1, M^2)$-trees operates as follows. Consider two leaves a_i, a_j in T, with labels $L'(a_i, T) = \langle L(a_i^0, T), L(a_i^1, T) \rangle$ and $L'(a_j, T) = \left\langle L(a_j^0, T), L(a_j^1, T) \right\rangle$. Suppose a_i and a_j are w-separated on level t in T. (Note that this implies that the corresponding leaves a_i^0 and a_j^0 are w_0-separated on level t of T_0, and a_i^1 and a_j^1 are w_1-separated on level t of T_1, where w_0 and w_1 are as defined earlier.)

Split the labels of a_i and a_j appropriately, and invoke the decoding function f twice, first to obtain $f(L(a_i^0, \hat{T}), L(a_j^0, \hat{T})) = \tilde{d}_{\hat{T}}(a_i^0, a_j^0)$ and then $f(L(a_i^1, \hat{T}), L(a_j^1, \hat{T})) = \tilde{d}_{\hat{T}}(a_i^1, a_j^1)$. By Lemma 2.16, these estimates allow us to extract the values of t, w_0 and w_1 in the process. From these three values it is easy to extract also k_0 and k_1, and hence also k and subsequently w. This allows us to compute the distance

$$d_T(a_i, a_j) \;=\; 2(\gamma_{t-1}(M^2) + w).$$

So we have obtained a $(1 + \epsilon)$-approximate (μ, M)-legal labeling scheme $\langle L', f' \rangle$ labeling any $(\mu - 1, M^2)$-tree with labels taken from $W(\mu, M, x)$. The lemma follows. ∎

Lemma 2.21 $|Q_\mu(M)| = M$.

Proof. Similarly to the proof of Lemma 2.15. ∎

Corollary 2.22 $g(\mu, M) \geqslant \sqrt{M} \cdot \sqrt{g(\mu-1, M^2)}$.

Proof. By Lemmas 2.19, 2.18, 2.20 and 2.21,

$$g(\mu, M)^2 \;\geqslant\; |\mathcal{W}| \;\geqslant\; M \cdot g(\mu-1, M^2)\ . \qquad \blacksquare$$

Lemma 2.23 $g(\mu, M) \geqslant M^{\mu/2}$.

Proof. By induction on μ. For $\mu = 1$, we have $|W(\mu, M, x)| \geqslant 1$, and so $|\mathcal{W}| \geqslant M$. On the other hand, by Lemma 2.18 we have $|\mathcal{W}| \leqslant g(1, M)^2$. Hence $g(1, M) \geqslant \sqrt{M}$, as claimed.

Assuming the claim for $\mu - 1$, we get by Corollary 2.22 and the inductive hypothesis,

$$g(\mu, M) \;\geqslant\; \sqrt{M}\sqrt{g(\mu-1, M^2)} \;\geqslant\; \sqrt{M}\sqrt{(M^2)^{(\mu-1)/2}} \;=\; \sqrt{M}\sqrt{M^{\mu-1}} \;=\; M^{\mu/2}\ . \qquad \blacksquare$$

This allows us to conclude with our two lower bounds. First let us consider weighted trees. Fix $\epsilon = 1$.

Theorem 2.24 *For the family $\mathcal{T}_{Z,n}$ of binary n-node trees with weights from the range $[0, Z)$, for sufficiently large n and $Z = \Omega(n^{\log n})$, $\ell_{(2,0)}(\mathcal{T}_{Z,n}) \geqslant \frac{1}{4}(\log n - 2)\log\log Z$.*

Proof. By Lemma 2.23, for the class $\mathcal{C}(\hat{\mu}, M)$ we have $\ell_{(2,0)}(\mathcal{C}(\hat{\mu}, M)) \geqslant \frac{\hat{\mu}}{2} \cdot \log M$. The maximum weight is $Z = 3^{2\hat{\mu}M-1}$. Assuming n, M are large enough (say, $n \geqslant 16$ and $M \geqslant 18\log n$), we get

$$\begin{aligned}
\ell_{(2,0)}(\mathcal{T}_{Z,n}) \;&\geqslant\; \frac{1}{2} \cdot \log\left(\frac{n}{3}\right) \cdot \log\left(\frac{1}{2\hat{\mu}} \cdot \left(\frac{\log Z}{\log 3} + 1\right)\right) \\
&\geqslant\; \frac{1}{2}(\log n - 2)(\log\log Z - \log\log n - 2) \\
&\geqslant\; \frac{1}{4}(\log n - 2)\log\log Z,
\end{aligned}$$

yielding the theorem. ∎

The above result can be extended to any constant approximation ratio $s > 2$, yielding the following.

Corollary 2.25 *For the family $\mathcal{T}_{Z,n}$ of binary n-node trees with weights from the range $[0, Z)$, for sufficiently large n and $Z = \Omega(n^{\log n})$, and for every constant $s \geqslant 2$, $\ell_{(s,0)}(\mathcal{T}_{Z,n}) = \Omega(\log n \cdot \log\log Z)$.*

For the lower bound on unweighted trees, fix $M = \lceil \log n \rceil$ and $\epsilon = 1/\log n$.

Theorem 2.26 *For the family $\mathcal{T}_n$ of unweighted binary n-node trees,*

$$\ell_{(1+6/\log n,0)}(\mathcal{T}_n) = \Omega(\log n \cdot \log\log n) \ .$$

Proof. Consider the family $\mathcal{T}'_N$ of binary weighted N-node trees $\mathcal{C}(\hat{\mu}, M)$ as defined above. By Lemma 2.23,

$$\ell_{(1+1/\log N,0)}(\mathcal{T}'_N) \ \geqslant \ \frac{\hat{\mu}}{2} \cdot \log M \ = \ \frac{1}{2} \cdot \log\left(\frac{N}{3}\right) \cdot \log\log N. \tag{2}$$

The maximum weight in any tree of $\mathcal{T}'_N$ is

$$\begin{aligned} Z &= \left\lceil A \cdot \beta^{2\hat{\mu}M-1} \right\rceil \ = \ O\left(\log^2 N \cdot \left(1 + \frac{2}{\log N}\right)^{(2\log(N/3)-1)\log N - 1}\right) \\ &= O(\log^2 N \cdot e^{4\log(N/3)}) \ = \ O(\log^2 N \cdot N^4). \end{aligned}$$

A $(\hat{\mu}, M)$-tree T' with integer edge weights $w(e) \geqslant 1$ can be transformed into an unweighted tree $T = \varphi(T')$ by replacing each edge e of weight w with a path of w (unweighted) edges. Clearly, any distance labeling scheme for the leaves of T will apply equally well to the leaves of T'. Hence, denoting the collection of transformed trees by $\varphi(\mathcal{T}'_N) = \{\varphi(T') \mid T' \in \mathcal{T}'_N\}$, and letting $t(N)$ be the maximum number of nodes in any unweighted tree in $\varphi(\mathcal{T}'_N)$, we have

$$\ell_{(1+1/\log N,0)}(\varphi(\mathcal{T}'_N)) \ \geqslant \ \ell_{(1+1/\log N,0)}(\mathcal{T}'_N) \tag{3}$$

Noting that

$$t(N) \ \leqslant \ 3 \cdot 2^{\hat{\mu}} \cdot Z \ = \ O(N \cdot \log^2 N \cdot N^4) \ = \ O(N^5 \cdot \log^2 N),$$

it follows that taking $N = n^{1/6}$ for sufficiently large n, every tree $T \in \varphi(\mathcal{T}'_N)$ has at most n nodes. Hence by Inequalities (2) and (3),

$$\begin{aligned} \ell_{(1+6/\log n,0)}(\mathcal{T}_{Z,n}) &\geqslant \ \ell_{(1+1/\log N,0)}(\varphi(\mathcal{T}'_N) \ \geqslant \ \ell_{(1+1/\log N,0)}(\mathcal{T}'_N) \\ &= \ \Omega(\log N \cdot \log\log N) \ = \ \Omega(\log n \cdot \log\log n), \end{aligned}$$

yielding the theorem. ∎

3 Additive Approximate Schemes

3.1 A Scheme for Bounded Chordality Graphs

A graph G is *k-chordal* if it does not contain any chordless cycles of length larger than k (where a *chord* is an edge joining two non-neighbors of the cycle). The *chordality* of G is the smallest integer $k \geqslant 2$ such that G is k-chordal. We adopt the convention that trees have chordality 2. *Chordal* graphs are exactly 3-chordal graphs.

We use the notion of tree-decomposition used by Roberston and Seymour in their work on graphs minors [RS86]. A *tree-decomposition* of a graph G is a tree T whose nodes are subsets of $V(G)$, such that

1. $\bigcup_{X \in V(T)} X = V(G)$;
2. for all $\{u, v\} \in E(G)$, there exists $X \in V(T)$ such that $u, v \in X$; and
3. for all $X, Y, Z \in V(T)$, if Y is on the path from X to Z in T, then $X \cap Z \subseteq Y$.

For every $S \subseteq V(G)$, denote $\mathrm{diam}_G(S) = \max_{u,v \in S} d_G(u, v)$ and $d_G(u, S) = \min_{v \in S} d_G(u, v)$, and denote the diameter of G by $\mathrm{diam}(G) = \mathrm{diam}_G(V(G))$. We denote by $G \setminus S$ the subgraph of G induced by the set of nodes $V(G) \setminus S$. Let $N(x)$ denote the set of neighbors of x (excluding x itself). Two nodes $x, y \in V(G)$ are *twins* if $N(x) \setminus \{y\} = N(y) \setminus \{x\}$.

S is a *separator* of G if $G \setminus S$ is composed of two or more connected components. Moreover, S is said to be *minimal* if every proper subset of S if not a separator of G. Given $x, y \in V(G)$, S is an *x, y-separator* if x and y belongs to two distinct connected components in $G \setminus S$. Furthermore, S is an *x-separator* if there exists some $y \in V(G)$ such that S is an x, y-separator. We have the following.

Lemma 3.1 *For every k-chordal graph G and minimal x, y-separator S,* $\mathrm{diam}_G(S) \leqslant k/2$.

Proof. Let C_x and C_y be the two connected components of $G \setminus S$ containing x and y respectively. Let $u, v \in S$, and let us bound $d_G(u, v)$. Without loss of generality choose u, v non-adjacent. We remark that u has one neighbor in C_x, say u_x, and one in C_y, say u_y (otherwise u could be removed from S, and S would be not minimal). Similarly, choose $v_x \in N(v) \cap V(C_x)$, and $v_y \in N(v) \cap V(C_y)$. Since C_x is connected, there exists a chordless path P_x between u_x and v_x, contained in its entirety in C_x. It follows that between u and v there exists a chordless path contained entirely in C_x except for its extremities, u and v (note that if P_x intersects $N(u)$ several times and creates some chords, then one can update u_x, selecting a new node in $V(P_x) \cap N(u)$, and skipping any chord). Similarly, there exists a chordless path between u and v contained entirely in C_y. These two paths combined form a chordless cycle, since u and v are not neighbors and since due to the separator S there is no edge between C_x and C_y. Because the length of this cycle is at most k, $d_G(u, v) \leqslant k/2$, completing the proof. ∎

In [BB98] it is shown,

Lemma 3.2 ([BB98]) *In every non-clique graph G, there exists a node $x \in V(G)$ such that $N(x)$ contains a minimal x-separator S and such that any $u \in N(x) \setminus S$ is a twin of x. Moreover, x and S can be found in polynomial-time.*

Theorem 3.3 *For every k-chordal graph G, there exists a tree-decomposition T such that $|V(T)| \leqslant |V(G)|$, and such that, for every $X \in V(T)$, $\mathrm{diam}_G(X) \leqslant k/2$. Moreover, T is polynomial-time constructible.*

Proof. We consider the following procedure for constructing a tree-decomposition T for a given k-chordal graph G.

<u>NICE-TREE</u>(G)

Input: a k-chordal graph G
Output: a tree-decomposition T

1. If G is a clique, then $V(T) \leftarrow V(G)$, $E(T) \leftarrow \varnothing$, and return T.
2. Select $x \in V(G)$ and $S \subseteq N(x)$ satisfying Lemma 3.2.
3. Set $X \leftarrow N(x) \cup \{x\}$.
4. Set $G' \leftarrow G \setminus (X \setminus S)$, and $E(G') \leftarrow E(G') \cup S^2$.
5. $T' \leftarrow$ NICE-TREE(G').
6. Let $X' \in V(T')$ such that $S \subseteq X'$.
7. Set $V(T) \leftarrow V(T') \cup \{X\}$, $E(T) \leftarrow E(T') \cup \{(X, X')\}$, and return T.

Let G_0 denote the initial graph G. The terminal case of Step 1 is trivial, so we can run Steps 2, 3 and 4. Note that in Step 4, we add all the edges in S and remove the connected component of x. Hence G' is composed of an induced subgraph of G, eventually with more edges. Therefore, at each call, S is a minimal x-separator of G_0 as well, and the chordality of G' is not greater than the chordality of G, thus of G_0. By Lemma 3.1, $\mathrm{diam}_{G_0}(S) \leqslant k/2$.

Consider Step 5 and consider T', the resulting tree of the call NICE-TREE(G'). Assume that, for every $X' \in V(T')$, $\mathrm{diam}_{G_0}(X') \leqslant k/2$. Let us show that the property holds for T as well. Because Step 7, it suffices to prove that $\mathrm{diam}_{G_0}(X) \leqslant k/2$. Let $u, v \in X$. First note that if both $u, v \in S$, then $d_G(u, v) \leqslant \mathrm{diam}_{G_0}(S) \leqslant k/2$ and we are done.

So hereafter, assume without loss of generality that $u \notin S$. Hence u is either x or a twin of x, so in particular it neighbors every node in $N(x)$. We now argue that $\{u, v\} \in E(G)$. Indeed, if $v \in S$, then $v \in N(x)$, hence it neighbors u; and if $v \in X \setminus S$, then by Lemma 3.2, v too is a twin of x, and thus u, v are neighbors in G.

Now, by induction on the recursive operation of the procedure, at each call of NICE-TREE, either the edge $\{u, v\}$ in G was also an edge in G_0, and thus $d_{G_0}(u, v) = 1$; or $\{u, v\}$ was not an edge in G_0, and thus it belongs to some previous minimal separator S' and we can conclude that $d_{G_0}(u, v) \leqslant \operatorname{diam}_{G_0}(S') \leqslant k/2$.

Due to the Rule 2 in the tree-decomposition definition, the node $X' \in V(T')$ in Step 6 exists, assuming T' is a tree-decomposition. So, by induction, merging the component X to T', we can form a tree-decomposition of G_0 as done in Step 7, the rules 1, 2, and 3 being clearly satisfied. Finally, the decomposition takes a polynomial time, and $|V(T)| \leqslant |V(G)|$ since at least one node ($X \setminus S$ contains at least x) is removed from G at each call in Step 4. ∎

Theorem 3.4 *There exists a $\lfloor k/2 \rfloor$-additive distance labeling scheme with labels of size $O(\log n \log D)$ for the family of k-chordal graphs with n nodes and (weighted) diameter D. Moreover, the scheme is polynomial-time constructible, and the distance decoder has $O(1)$-time complexity.*

Proof. We use Procedure NICE-TREE to decompose G. By Theorem 3.3, $|V(T)| \leqslant n$. We apply the scheme of Theorem 2.5 to T. Let $U \in V(T)$. U has a separator sequence $X_1, \ldots, X_h$ of nodes of T, $h \leqslant \log n$. The lowest common ancestor X_i can be determined in constant time storing a sequence C on $O(\log n)$ bits only. As done in Theorem 2.5. The label of any yet unlabeled node $u \in U$ consists of C and of the sequence of distances $d_G(u, X_1), \ldots, d_G(u, X_h)$. The length of the label is bounded by $h \lceil \log D \rceil + O(\log n) = O(\log n \log D)$ bits, where D is an upper bound on the (weighted) distances.

The distance decoder returns $d_G(u, X_i) + d_G(v, X_i) + \lfloor k/2 \rfloor$, where X_i is the separator between u and v. Since X_i is a separator, $d_G(u, v) \geqslant d_G(u, X_i) + d_G(v, X_i)$, and since $\operatorname{diam}_G(X_i) \leqslant k/2$, we have $d_G(u, X_i) + d_G(v, X_i) + \lfloor k/2 \rfloor \geqslant d_G(u, v)$. ∎

Corollary 3.5 *There exists a 1-additive distance labeling scheme with labels of size $O(\log^2 n)$ for the family of chordal graphs with n nodes. Moreover the scheme is polynomial-time constructible, and the distance decoder has $O(1)$-time complexity.*

The previous bound is optimal with respect to both the approximation ratio and the label size. Indeed, as seen later in Section 3.3, every 1-additive distance labeling scheme on the family of trees (that are chordal) requires some labels of size $\Omega(\log^2 n)$. Moreover, every exact distance labeling scheme on chordal graphs of diameter 3 requires label of size $\Omega(n)$ (cf. Theorem 3.20).

3.2 Dominating Pair

Our next goal is to show that in certain cases, Theorem 3.4 can be improved upon. A pair of nodes $\{x, y\}$ in a graph G is called a *dominating pair* if for every path P in G between

x and y, and for every $u \in V(G)$, there exists a node of P at distance at most 1 of u (we say that P dominates u).

Consider any graph G having a dominating pair $\{x, y\}$, and let $\delta = d_G(x, y)$. The distance labeling scheme for graphs having a dominating pair is based on the decomposition by distance layers from node x. Let us define $L_t = \{u \in V(G) \mid d_G(x, u) = t\}$, for every $t \geqslant 0$. Note that for every $0 < t < \delta$, L_t is a x, y-separator. Let S_t denote a minimal x, y-separator in L_t, for every $0 < t < \delta$. We set $S_0 = \{x\}$, $S_\delta = \{y\}$ and $S_{\delta+1} = \varnothing$.

An *x, y-path* is a loop-less path between x and y.

Claim 3.6 *Let C_x and C_y be the connected components of $G \setminus S_t$, for $0 < t < \delta$, containing respectively x and y. For any node $z \in S_t$, there exists an x, y-path contained in $C_x \cup C_y \cup \{z\}$, composed of a shortest x, z-path and of an induced z, y-path (i.e., a chordless z, y-path).*

Proof. Since S_t is a minimal separator, any node $z \in S$ has a neighbor in C_x (resp. C_y) the connected component of $G \setminus S_t$ containing x (resp. y). Therefore there exists an induced x, z-path P_{xz} (resp. z, y-path P_{zy}) such that $P_{xz} \setminus C_x = \{z\}$ (resp. $P_{zy} \setminus C_y = \{z\}$). Moreover since the nodes of S_t are equidistant from x, P_{xz} can be chosen as a shortest path. ∎

Hereafter, an x, y-path satisfying Claim 3.6 for some $z \in S_t$ is called an *x, z, y-path*. Given a u, v-path P_{uv} and a v, w-path P_{vw}, we denote by $P_{uv} \cdot P_{vw}$ the u, w-path obtained by joining P_{uv} and P_{vw}.

Claim 3.7 *Let t, t' such that $0 < t \leqslant t' < \delta$. Let P be a x, z, y-path with $z \in S_t$. Then, every $w' \in L_{t'} \setminus S_{t'}$ has no neighbors in $P \cap L_{t'+1}$.*

Proof. Since $w \in L_{t'} \setminus S_{t'}$, there exists a shortest x, w-path P_{xw} that avoids $S_{t'}$. Assume w has a neighor $w' \in P \cap L_{t'+1}$. Let $P_{w'y}$ be the w', y-path subpath of P. The path $P_{xw} \cdot \{w, w'\} \cdot P_{w'y}$ is a x, y-path that does not cut $S_{t'}$: contradiction, $S_{t'}$ is a minimal x, y-separator. ∎

Claim 3.8 *For every $t \geqslant 1$, and every $w \in L_t$, w has a neighbor in S_{t-1}.*

Proof. For the case $t = 1$, since $S_0 = \{x\}$ and $L_1 = N(x)$, the claim holds. For $t \geqslant \delta + 1$, since $\{x, y\}$ is a dominating pair, any shortest x, y-path dominates the graph, y has to dominates every node of L_t, if exists. Recall that $S_\delta = \{y\}$.

Consider P_{xw} a shortest x, w-path, and let $w_{t-1} \in P_{xw} \cap L_{t-1}$. If $w_{t-1} \in S_{t-1}$,we are done. If not, P_{xw} avoids S_{t-1}. Let P_{xy} be a shortest x, y-path (it is composed by one node per S_i's). It has to dominate w. So, let v be a neighbor of w belonging to $P_{x,y}$. If $v \in S_{t-1}$, we are done. So, $v \in S_t \cup S_{t+1}$. Let P_{vy} be the v,y-path subpath of P_{xy}. The path $P_{xw} \cdot \{w,v\} \cdot P_{vy}$ is an x,y-path that avoids S_{t-1}: contradiction, S_{t-1} is a x,y-separator. Therefore w has a neighbor in S_{t-1}. ∎

Let us now consider two distinct nodes u and v, and let $t_u = d_G(x,u)$ and let where $t_v = d_G(x,v)$. W.l.o.g., assume that $t_u \leqslant t_v$. The next claims give approximations of $d_G(u,v)$ depending on the respective positions of u and v in the distance layer decomposition.

Claim 3.9 *If* $t_u = 0$, *then* $d_G(u,v) = t_v$.

Claim 3.10 *If* $0 < t_u = t_v \leqslant \delta + 1$, *then* $1 \leqslant d_G(u,v) \leqslant 2$.

Proof. Since $u \neq v$, $d_G(u,v) \geqslant 1$. Let $t = t_u = t_v$. If $t \geqslant \delta + 1$, then both u and v have to be adjacent to y (otherwise any shortest x,y-path would not dominate the graph), and thus $d_G(u,v) \leqslant 2$. So, assume $0 < t \leqslant \delta$, and u and v are not adjacent.

Assume first that $v \in S_t$. Note that $t < \delta$, otherwise $v = y$ and v would be adjacent to u. Let P be a x,v,y-path. Let v_{t-1} be the neighbor of v in $P \cap L_{t-1}$. P has to dominate u. By Claim 3.7, u has no neighbor in $P \cap L_{t'+1}$ for $t' \geqslant t$. Thus u is adjacent to v_{t-1}, and $d_G(u,v) = 2$.

So, assume now that $u,v \in L_t \setminus S_t$. By Claim 3.8, u and v have both a neighbor in S_{t-1}, namely u_{t-1} and v_{t-1}. If they have a common neighbor in S_{t-1}, then $d_G(u,v) = 2$. Assume u and v have no common neighbor in S_{t-1}. Let $P = P_{xu_{t-1}} \cdot P_{u_{t-1}y}$ be a x,u_{t-1},y-path. P has to dominate v that is not adjacent to u_{t-1} and u. Let $[w_1,\dots,w_s]$ be the subpath of P induced by L_t. If $t = \delta$, then $d_G(u,v) = 2$. Thus $t < \delta$. By Claim 3.7, v has no neighbors in $P \cap L_{t+1}$, thus either $v = w_j$ or is adjacent to some w_j, $j \in \{1,\dots,s\}$. Let us pick j as the largest as possible. Let $P_{xv_{t-1}}$ be a shortest x,v_{t-1}-path. Consider $Q = P_{xv_{t-1}} \cdot [v_{t-1}, v, w_j, \dots, w_s, \dots, y]$ a x,y-path (noting that $w_j = v$ is possible). Q has to dominates u. By Claim 3.7, u has no neighbors in $P \cap L_{t+1}$, thus either $u = w_k$ or is adjacent to some w_k, $k \in \{j,\dots,s\}$. Finaly, consider $R = P_{xu_{t-1}} \cdot [u_{t-1}, u, w_k, \dots, w_s, \dots, y]$ a x,y-path (noting that $w_k = u$ is possible). R has to dominates v, and by Claim 3.7, either $v = w_r$ or is adjacent to some $w_r \in \{k,\dots,s\}$. Since j was chosen as large as possible, it implies that $r = k = j$, and thus $d_G(u,v) = 2$. ∎

Corollary 3.11 *If* $0 < t_u < t_v \leqslant \delta + 1$, *then* $t_v - t_u \leqslant d_G(u,v) \leqslant t_v - t_u + 2$.

Proof. Since $v \in L_{t_v}$ there exists a shortest x,v-path P that uses one node per L_i for $i \leqslant t_v$. Therefore $d_G(v, S_{t_u}) = t_v - t_u$. Let w be the node of $P \cap L_{t_u}$. Since by Claim 3.10 $d_G(u,w) \leqslant 2$, it follows that $d_G(u,v) \leqslant d_G(v,w) + 2 = t_v - t_u + 2$. ∎

Theorem 3.12 *Let G be a graph of diameter D with a dominating pair. Then G has a 2-additive distance labeling with labels of size $\log D + O(1)$. Moreover, the distance decoder has $O(1)$-time complexity.*

Proof. Let $\{x, y\}$ be a dominating pair of G. Let $L(u, G) = t_u$ be the label of any node u of G, where $t_u = d_G(x, u)$. The approximate distance $\tilde{d}(u, v)$ between u and v can be computed as follows:

$$\tilde{d}(u,v) \;=\; \begin{cases} t_v, & \text{if } t_u = 0, \\ t_v - t_u + 2, & \text{if } t_u \neq 0. \end{cases}$$

By Claim 3.9 and 3.10, and by Corollary 3.11, $\tilde{d}(u, v)$ is a 2-additive estimator on $d_G(u, v)$. The distance decoder has $O(1)$ time complexity. And since $t_u \in \{0, \ldots, D\}$, the length of each label $L(u, G)$ is of size at most $\lceil \log(D + 1) \rceil$. ∎

An *asteroidal triple* of a graph G is an independent triple of nodes, each two of which are joined by a path of G that avoids the neighborhood of the third node [LB62]. Graphs without asteroidal triple are termed *AT-free graphs*. This class includes, in particular, the classes of interval, permutation, bounded tolerance and co-comparability graphs (see [Gol80]). It is well-known that AT-free graphs have a dominating pair [COS97] that can be founded in linear time. We thus have the following.

Corollary 3.13 *There exists a 2-additive distance labeling scheme with labels of size $\log D + O(1)$ for the family of AT-free graphs of diameter D. Moreover, the scheme is polynomial-time constructible, and the distance decoder has $O(1)$-time complexity.*

To get a 1-additive scheme, we examine in more details the situation described by 3.11: $t_u < t_v$, i.e., $d_G(x, u) < d_G(x, v)$. Indeed, thank to Claim 3.9, we have already a 1-additive scheme for $t_u = t_v$.

Claim 3.14 *If $1 < t_u + 1 < t_v \leqslant \delta + 1$ and $u \notin S_{t_u}$, then $t_v - t_u + 1 \leqslant d_G(u, v) \leqslant t_v - t_u + 2$.*

Proof. Clearly the edge $\{u, v\}$ cannot exist. Let us show that S_{t_u} is a u, v-separator.

Let us assume there exists a u, v-path P such that $P \cap S_{t_u} = \varnothing$. Since $u \in L_{t_u}$, there exist a shortest x, u-path P' such that $P' \cap S_{t_u} = \varnothing$. By Claim 3.8, v has a neighbor $v_{t_v-1} \in S_{t_v-1}$. Notice that $v_t \notin S_{t_u}$. By Claim 3.6, there exists a x, v_t, y path that is the concatenation of two paths P_{xv_t} and P_{v_ty}. Notice that $P_{v_ty} \cap S_{t_u} = \varnothing$. Now the x, y-path $P' \cdot P \cdot P_{v_ty}$ is not cut by S_{t_u}: contradiction.

Therefore there exists a node $z \in S_{t_u}$ such that $d_G(u, v) = d_G(u, z) + d_G(z, v)$. Note that $d_G(z, v) = t_v - t_u$. Since u and z both belong to L_{t_u}, $1 \leqslant d_G(u, z) \leqslant 2$. Thus

$t_v - t_u + 1 \leqslant d_G(u, v) \leqslant t_v - t_u + 2.$ ∎

We say that u *dominates* subset of nodes S, if every node of S is a neighbor of u. In the next two claims, we will examine the case where u dominates S_{t_u}, and the case where u belongs to a chordless cycle of length larger or equal to 4.

Claim 3.15 *If $u \notin S_{t_u}$ and u dominates S_{t_u}, then for any node $v \in L_{t_u+1}$, $1 \leqslant d_G(u, v) \leqslant 2$.*

Proof. Since $u \neq v$, $d_G(u, v) \geqslant 1$. By Claim 3.8, v has a neighbor $v_{t_u} \in S_{t_u}$. Since u dominates S_{t_u}, v_{t_u} is a common neighbor of u and v. Thus $1 \leqslant d_G(u, v) \leqslant 2$. ∎

Claim 3.16 *Assume that $t_u > 0$, $u \notin S_{t_u}$, and u does not dominate S_{t_u}. If u does not belong to a chordless cycle C_k, $k \geqslant 4$, then there is no $v \in L_{t_u+1}$ and $w \in L_{t_u+1}$ such that $d_G(u, v) = 3$ and $d_G(u, w) = 1$.*

Proof. Let us first show that if u has a neighbor $w \in L_{t_u+1} \setminus S_{t_u+1}$, then w has to dominate S_{t_u}. Assume there exists $w' \in S_{t_u}$ not adjacent to w. By Claim 3.6, there exists a x, w', y-path $P = P_{xw'} \cdot P_{w'y}$ that has to dominate w. Since $u \in L_{t_u}$ there exists a x, u-path P' such that $P' \cap S_{t_u} = \varnothing$. If w is adjacent to some node z of $P_{w_{t_u}y}$, then the path $P' \cdot [w, z, \ldots, y]$ is a x, y-path that is not cut by S_{t_u}: contradiction, thereby w cannot be dominated by $P_{w'y}$. Because of distance arguments w cannot be dominated by $P_{xw'}$. In other words the x, y-path P does not dominate w: contradiction, w has to dominate S_{t_u}.

Let us now prove that if u does not belong to any cycle of length $k \geqslant 4$, it cannot exist in L_{t+1} a node w such that $d_G(u, w) = 1$ and a node v such that $d_G(u, v) = 3$. Let us assume it is the case. By Claim 3.8, v has a neighbor $v_{t_u} \in S_{t_u}$. Since $d_G(u, v) = 3$, u and v_{t_u} are not adjacent. Since w dominates S_{t_u}, (v_{t_u}, w) is an edge. Since u and v_{t_u} belong to L_{t_u}, there exists shortest paths $P = [x, \ldots, u]$ and $P' = [x, \ldots, v_{t_u}]$. Now $\mathcal{C} = P \cdot P' \cdot [w]$ is a cycle of length more than $k \geqslant 4$. Since $N(w) \cap \mathcal{C} = \{u, v_{t_u}\}$ and $\{u, v_{t_u}\}$ is not an edge, there exists a chordless cycle of length more than $k \geqslant 4$ that contains node u: contradiction. ∎

Claim 3.17 *Assume that $0 < t_u < t_v \leqslant \delta + 1$, and $u \in S_{t_u}$. If u does not belong to a chordless cycle C_k, $k \geqslant 4$, then $t_v - t_u \leqslant d_G(u, v) \leqslant t_v - t_u + 1$.*

Proof. By Claim 3.8, v has a neighbor $v_{t_v-1} \in S_{t_v-1}$. If $t_v = t_u + 1$, then let us rename v_{t_v-1} by v_{t_u}. Otherwise there exists a shortest x, v-path P such that $S_{t_u} \cap P = \{v_{t_u}\}$. Moreover we have $d_G(v_{t_u}, v) = t_v - t_u$. If $\{u, v_{t_u}\}$ is not an edge, then the minimal separator S_{t_u} is not complete. It implies there exists a C_k with $k \geqslant 4$ containing u: contradiction, thereby $t_v - t_u \leqslant d_G(u,v) \leqslant t_v - t_u + 1$. ∎

Using the presented pool of claims, it is possible to strengthen Theorem 3.12 in some cases. Indeed these claims show that the worst-case of the 2-additive estimator of $d_G(u,v)$ is reached only whenever u (the closest to x) belongs to a chordless cycle of length 4 or more. By the way, storing some more bits of information, for some interesting classes of graphs we can propose a 1-additive distance labeling scheme. So, all we have to know is the four following bits:

- whether u belongs to the minimal separator S_{t_u},
- whether u belongs to an induced cycle C_k, $k \geqslant 4$,
- whether u dominates the minimal separator S_{t_u},
- whether u is adjacent to some node of the distance layer L_{t_u+1}.

These bits will be respectively named s_u, c_u, d_u and a_u. (They are set to 1 if the situation arises, and set to 0 otherwise.) So let us define the distance labeling scheme $\mathcal{L}^* = \langle L, f\rangle$ of G such that the label of any node u is $L(u,G) = (t_u, s_u, d_u, c_u, a_u)$ where $t_u = d_G(x,u)$ and s_u, d_u, c_u and a_u as the bits informations previously mentioned. And for any two distinct nodes u and v such that $t_u \leqslant t_v$, the distance decoder $f(L(u,G), L(v,G))$ returns $\tilde{d}(u,v)$ defined as follows:

$$\tilde{d}(u,v) = \begin{cases} t_v, & \text{if } t_u = 0, \\ 2, & \text{if } t_u = t_v, \\ t_v - t_u + 1, & \text{if } s_u = 1 \text{ and } t_u < t_v \text{ and } c_u = 0, \qquad (1) \\ t_v - t_u + 2, & \text{if } s_u = 0 \text{ and } t_u + 1 < t_v, \\ 2, & \text{if } s_u = 0,\ t_u + 1 = t_v \text{ and } d_u = 1, \\ 2, & \text{if } s_u = 0,\ t_u + 1 = t_v,\ d_u = 0,\ c_u = 0 \text{ and } a_u = 1, \qquad (2) \\ 3, & \text{if } s_u = 0,\ t_u + 1 = t_v,\ d_u = 0,\ c_u = 0 \text{ and } a_u = 0. \qquad (3) \end{cases}$$

Clearly the labels are of size $\log D + O(1)$ and the distance decoder has $O(1)$-time complexity.

Lemma 3.18 *Let G be a graph that has a dominating pair $\{x,y\}$. For any two nodes u and v of G, with $d_G(x,u) \leqslant d_G(x,v)$, if u does not belong two a chordless cycle C_k, $k \geqslant 4$, then $d_G(u,v) \leqslant \tilde{d}(u,v) \leqslant d_G(u,v) + 1$.*

Proof. If $t_u = 0$, then Claim 3.9 proves the results. If u and v are equidistant from x, we can apply Claim 3.10.

Let us assume now that $t_u < t_v$. If u belongs the the minimal separator S_{t_u} (i.e., then since u does not belong to any chordless cycle of length 4 or more), Claim 3.17 gives the answer.

Let us assume now that u does not belong to S_{t_u}. If $t_u + 1 < t_v$, the result hold by Claim 3.14. We now have to examine the case where u and v belongs to consecutive distance layer L_{t_u} and L_{t_u+1}. By Claim 3.15, if $d_u = 1$, the property holds. Otherwise by Claim 3.16, whenever u u does not belong to any chordless cycle of length 4 or more, all we have to know is whether u as a neighbor in L_{t_u+1}. If so returning 2 gives the right estimator, otherwise we have to return 3. ∎

Since interval graphs are 3-chordal AT-free graphs, the following corollary holds:

Corollary 3.19 *There exists a* 1*-additive distance labeling scheme with labels of size* $\log D + O(1)$ *for interval graphs of diameter* D*. Moreover, the scheme is polynomial-time constructible, and the distance decoder has* $O(1)$*-time complexity.*

Note that the distance labeling scheme $\mathcal{L}^*$ gives also a 1-additive scheme for graphs having a dominating pair, and that are neither interval nor AT-free graphs. For instance consider the family composed of (connected) induced subgraphs of the graph depicted on Fig. 3.

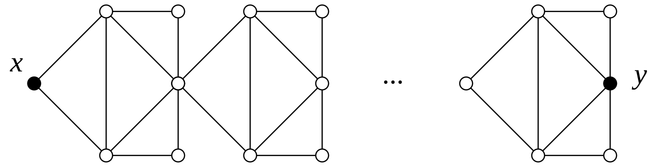

Figure 3: A non AT-free graph having a dominating pair.

3.3 Lower Bounds for Additive Schemes

In this section we establish some lower bounds on r-additive distance labeling schemes. For general graphs we show that labels of size $\Omega(\sqrt{n/r})$ are required in the worst-case (the lower bound can be improved to $\Theta(n)$ for $r < 2$), and that every r-additive distance labeling on k-chordal graphs must use labels of length $\Omega(\log^2 n)$, for every constant $r \geqslant 0$. Moreover, there is no exact distance labeling scheme with labels shorter than $\Omega(n)$ bits, proving that the scheme presented in Section 3.1 is optimal (cf. Theorem 3.4 and Corollary 3.5).

Let us now show that the labeling scheme of Section 3.1 for bounded chordality graphs is optimal in terms of the length of its labels. First, let us show that there is no exact distance labeling scheme for chordal graphs using "short" labels. Consider the family $\mathcal{S}_n$ of connected *split* graphs, namely, all the n-node graphs composed of a clique C, and of an independent set I of $n - |C|$ nodes, such that each node of I is connected to at least one node of C. Let $\mathcal{C}_{n,k}$ be the class of connected n-node k-chordal graphs. Clearly, $\mathcal{S}_n$ is a subclass of $\mathcal{C}_{n,3}$ (because every cycle of 4 or more nodes has a chord belonging to C) and $\mathcal{S}_n$ is connected (in fact, it contains only graphs of diameter at most three).

Recall that $\overline{\ell}_{(s,r)}(\cdot)$ stands for the total label length (cf. Section 2.3), and that $\ell_{(s,r)}(\cdot)$ and $\overline{\ell}_{(s,r)}(\cdot)$ as well are related to the non-distinct label model.

Theorem 3.20 *For every $k \geqslant 3$ and $s < 2$, $\ell_{(s,0)}(\mathcal{C}_{n,k}) \geqslant \ell_{(s,0)}(\mathcal{S}_n) \geqslant n/4 - O(1)$ and $\overline{\ell}_{(s,0)}(\mathcal{C}_{n,k}) \geqslant \overline{\ell}_{(s,0)}(\mathcal{S}_n) \geqslant n^2/4 - O(n \log n)$. On the other hand, the family $\mathcal{S}_n$ supports a 1-additive distance labeling scheme with 1 bit labels (or $\lceil \log n \rceil + 1$ bit label, if we insist on distinct labels).*

Proof. Since $\mathcal{S}_n \subseteq \mathcal{C}_{n,3}$ and $\mathcal{C}_{n,k} \subseteq \mathcal{C}_{n,k+1}$, it suffices to prove the lower bound for $\mathcal{S}_n$ to prove a lower bound for all k-chordal graphs. Note that $|\mathcal{S}_n| \geqslant 2^{(n-|C|)\cdot(|C|-1)} \geqslant 2^{n^2/4-O(n)}$ (choosing $|C| = \lceil n/2 \rceil$).

Let $s < 2$. Two graphs $G, H \in \mathcal{S}_n$ differ by an edge joining a node of C with a node of I. Therefore, there exists a pair $(u,v) \in V_n^2$ such that $d_G(u,v) = 1$ whereas $d_H(u,v) = 2$, or $d_G(u,v) = 2$ whereas $d_H(u,v) = 1$. Thus $\mathcal{S}_n$ is a $(V_n, 2)$-family. The result now follows from Lemma 2.9.

In order to label the nodes, we set $L(v,G) = 0$ if $v \in C$, and $L(v,G) = 1$ if $v \in I$. (If we insist on distinct labels to nodes, then the scheme must be modified to assign a unique identifier of size $\log n$ to each node.) The approximate distance is calculated for two nodes $u, v \in G$, with labels $L(u,G)$ and $L(v,G)$, as follows:

1. If $L(u,G) = L(v,G) = 0$, then return 1.
2. If $L(u,G) \neq L(v,G)$, then return 2.
3. If $L(u,G) = L(v,G) = 1$, then return 3.

Clearly, in the first case the answer is exact, and in the other two cases the answer in either exact or greater by 1 from the true distance. ∎

Similarly, we have the following for AT-free graphs.

Theorem 3.21 *Let $\mathcal{A}_n$ be the class of connected n-node AT-free graphs. For every $s < 2$, $\ell_{(s,0)}(\mathcal{A}_n) \geqslant n/4 - O(1)$ and $\overline{\ell}_{(s,0)}(\mathcal{A}_n) \geqslant n^2/4 - O(n \log n)$.*

Proof. Consider the family $\mathcal{S}'_n$ of graphs composed of split graphs, excepted that the nodes of the I set are completed by a clique. Then $\mathcal{S}'_n \subset \mathcal{A}_n$, because every triple of nodes contains at least a pair of adjacent nodes. The same conclusion as Theorem 3.20 holds in particular because $|\mathcal{S}'_n| = |\mathcal{S}_n|$. ∎

Now we will show a lower bound of $\Omega(\sqrt{n/r})$ on the label size for general graphs. We start with a simple observation.

Lemma 3.22 *For every $s \geqslant 1$, $r \geqslant 0$, and for every graph family $\mathcal{F}$, $\ell_{(s,r)}(\mathcal{F}) \geqslant \ell_{(s+r,0)}(\mathcal{F})$.*

Proof. It suffices to observe that for each distance[1] $d \geqslant 1$ and (s,r)-approximation $\tilde{d}$ of d, we have $d \leqslant \tilde{d} \leqslant s \cdot d + r \leqslant (s+r) \cdot d$. So any (s,r)-approximate distance labeling scheme on $\mathcal{F}$ is $(s+r,0)$-approximated as well. ∎

Thus, for general graphs, since $\ell_{(s,0)}(\mathcal{G}_n) = \Theta(n)$ for every $s < 3$ (cf. [GPPR01]), we have $\ell_{(1,r)}(\mathcal{G}_n) = \Theta(n)$ for every $r < 2$. We will extend this result later for larger r (cf. Corollary 3.25).

But first, we complete Theorem 3.5, by showing that even for $r > \lfloor k/2 \rfloor$, every r-additive scheme for k-chordal graphs requires labels of length $\Omega(\log^2 n)$.

Let $\mathcal{T}$ be the family of trees, and let $\mathcal{T}_n \subset \mathcal{T}$ denote n-node trees. Since $\mathcal{T}_n \subseteq \mathcal{C}_{n,k}$, it suffices to show that $\ell_{(1,r)}(\mathcal{T}_n) = \Omega(\log^2 n)$ for every constant r. For that, we need the following definitions and result. A *subdivision* of a graph G is a graph obtained from G by replacing each edge between x and y in G by a path of arbitrary length $\geqslant 1$ with extremities x and y. An infinite graph family $\mathcal{F}$ is *closed under subdivisions* if for every $G \in \mathcal{F}$, every subdivision of G belongs to $\mathcal{F}$.

Lemma 3.23 *Let $\mathcal{F}$ be a family of graphs closed under subdivisions, and let $\mathcal{F}_n^m \subset \mathcal{F}$ denote the subset of graphs with n nodes and m edges. Then for all $s \geqslant 1$ and $r, l \geqslant 0$, $\ell_{(s,r)}(\mathcal{F}_{n+lm}^{lm}) \geqslant \ell_{(s,\lfloor r/(l+1) \rfloor)}(\mathcal{F}_n^m)$.*

Proof. Let $\alpha = \ell_{(s,r)}(\mathcal{F}_{n+lm}^{lm})$, and let $\langle L, f \rangle$ be any (s,r)-approximate distance labeling scheme on $\mathcal{F}_{n+lm}^{lm}$ using labels of length at most α bits. Let $G \in \mathcal{F}_n^m$. Consider the graph G_l obtained from G by subdividing all its edges into path of length $l+1$, and label its new nodes arbitrarily with integers from $\{n+1, \ldots, n+lm\}$. $G_l \in \mathcal{F}_{n+lm}^{lm}$, thus for each node $u \in V_{n+lm}$, $L(u, G_l)$ is of size at most α, in particular for $u \in V_n \subset V_{n+lm}$.

We consider the distance labeling scheme $\langle L', f' \rangle$ on $\mathcal{F}_n^m$ defined as follows: $L'(u,G) = L(u, G_l)$, and $f'(L'(u,G), L'(v,G)) = \lfloor f(L(u,G_l), L(v,G_l))/(l+1) \rfloor$ for all $u, v \in V_n$. For simplicity, denote $\tilde{d}(u,v) = f(L(u,G_l), L(v,G_l))$ and $\tilde{d}'(u,v) = f'(L'(u,G), L'(v,G))$.

By definition of $\langle L, f \rangle$ we have, for all $u, v \in V_n$:

$$d_{G_l}(u,v) \ \leqslant\ \tilde{d}(u,v) \ \leqslant\ s \cdot d_{G_l}(u,v) + r \ .$$

Because the subdivision is applied on all the edges of G, $d_{G_l}(u,v) = (l+1) \cdot d_G(u,v)$. It follows that

$$d_G(u,v) \ \leqslant\ \left\lfloor \tilde{d}(u,v)/(l+1) \right\rfloor = \tilde{d}'(u,v) \ \leqslant\ s \cdot d_G(u,v) + \lfloor r/(l+1) \rfloor \ ,$$

hence $\langle L', f' \rangle$ is an $(s, \lfloor r/(l+1) \rfloor)$-approximate distance labeling for G. Moreover, the labels used for G are those used for G_l, thus $|L'(u,G)| = \alpha$, proving that $\ell_{(s,\lfloor r/(l+1) \rfloor)}(\mathcal{F}_n^m) \leqslant \alpha$.

[1]Recall that in the non-distinct label model, so $d = 0$ is not considered.

∎

Therefore, we have:

Corollary 3.24 *For every $r \geqslant 0$, $\ell_{(1,r)}(\mathcal{T}_n) = \Omega(\log^2(n/(r+1)))$, and thus, for every $k \geqslant 3$, $\ell_{(1,r)}(\mathcal{C}_{n,k}) = \Omega(\log^2(n/(r+1)))$.*

Proof. Rewrite Lemma 3.23 with $s = 1$, $r = l$, and $\mathcal{F} = \mathcal{T}$. We have

$$\ell_{(1,r)}(\mathcal{T}_{n+r(n-1)}) \ \geqslant \ \ell_{(1,0)}(\mathcal{T}_n) \ \geqslant \ \frac{1}{8}\log^2 n - O(\log n)$$

because $\lfloor r/(l+1) \rfloor = 0$, and by the result of [GPPR00] on binary n-node trees. Setting $N = n + r(n-1)$, we obtain, for every $r \geqslant 0$, $\ell_{(1,r)}(\mathcal{T}_N) = \Omega(\log^2(N/(r+1)))$. ∎

We can also improve the lower bound derived from Lemma 3.22 on r-additive schemes on general graphs.

Corollary 3.25 *$\ell_{(1,r)}(\mathcal{G}_n) = \Theta(n)$ for $r < 2$, and $\ell_{(1,r)}(\mathcal{G}_n) = \Omega(\sqrt{n/r})$, for every $r \geqslant 2$.*

Proof. We know that $\ell_{(1,0)}(\mathcal{G}_n) = O(n)$, and the result holds for every $r > 0$. Because $\ell_{(s,0)}(\mathcal{G}_n) = \Omega(n)$ for any $s < 3$, Lemma 3.22 gives $\ell_{(1,r)}(\mathcal{G}_n) = \Omega(n)$ for every $r < 2$.

Let $\mathcal{B}$ be the family of bounded degree graphs, and let $\mathcal{B}_n \subset \mathcal{B}$ denote the n-node graphs with maximum degree three. In [GPPR00], it is shown that $\ell_{(1,0)}(\mathcal{B}_n) = \Omega(n^{1/2})$. By Lemma 3.23,

$$\ell_{(1,r)}(\mathcal{B}_{n+1.5rn}) \ \geqslant \ \ell_{(1,0)}(\mathcal{B}_n) \ = \ \Omega(n^{1/2}) \ .$$

Setting $N = n + 1.5rn$, we obtain, for every $r \geqslant 0$, $\ell_{(1,r)}(\mathcal{B}_N) = \Omega((N/(r+1))^{1/2})$, that is $\Omega(\sqrt{N/r})$ for $r \geqslant 1$. ∎

4 Open problems

We conclude with the following open problems. Let $\mathcal{P}_n$ (respectively, $\mathcal{B}_n$, $\mathcal{I}_n$, $\mathcal{A}_n$) denote the families of n-node planar (resp., bounded-degree, interval, AT-free) graphs.

Problem 1 *Is there a constant s such that $\ell_{(s,0)}(\mathcal{P}_n) \leqslant \log^{O(1)} n$?*

Problem 2 *Determine the necessary label size for* exact *distance labeling for the above graph classes, i.e., find $\ell_{(1,0)}(\mathcal{P}_n)$, $\ell_{(1,0)}(\mathcal{B}_n)$ and $\ell_{(1,0)}(\mathcal{I}_n)$. The known bounds are:*
$\Omega(n^{1/3}) \leqslant \ell_{(1,0)}(\mathcal{P}_n) \leqslant O(n^{1/2}\log n)$,
$\Omega(n^{1/2}) \leqslant \ell_{(1,0)}(\mathcal{B}_n) \leqslant O(n)$,
$\Omega(\log n) \leqslant \ell_{(1,0)}(\mathcal{I}_n) \leqslant O(\log^2 n)$.

Problem 3 *Determine the necessary label size for 1-additive distance labeling scheme for the class $\mathcal{A}_n$, namely, establish $\ell_{(1,1)}(\mathcal{A}_n)$.*

Acknowledgment. We would like to thank Stéphane Pérennes for his help in developing the upper bound for trees, and Douglas West for his bibliographic support.

References

[BB98] Anne Berry and Jean-Paul Bordat. Separability generalizes dirac's theorem. *Discrete Applied Mathematics*, 84:43–53, 1998.

[COS97] Derek G. Corneil, Stephan Olariu, and Lorna Stewart. Asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 10(3):399–430, August 1997.

[ERS66] Paul Erdös, A. Rényi, and V. T. Sós. On a problem of graph theory. In *Studia Sci. Math. Hungar.*, volume 1, pages 215–234, 1966.

[FJ89] Greg N. Frederickson and Ravi Janardan. Efficient message routing in planar networks. *SIAM Journal on Computing*, 18(4):843–857, August 1989.

[FKS84] M. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, pages 538–544, 1984.

[FW93] Michael L. Fredman and Dan E. Willard. Surpassing the information theoric bound with fusion trees. *Journal of Computer and System Sciences*, 47:424–436, 1993.

[Gol80] Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, Harcourt Brace Jovanovich, Academic Press edition, 1980.

[GP72] Ronald Lewis Graham and H. O. Pollak. On embedding graphs in squashed cubes. *Lecture Notes in Mathematics*, 303:99–110, 1972.

[GPPR00] Cyril Gavoille, David Peleg, Stéphane Pérennes, and Ran Raz. Distance labeling in graphs. Research Report RR-1239-00, LaBRI, University of Bordeaux, 351, cours de la Libération, 33405 Talence Cedex, France, April 2000.

[GPPR01] Cyril Gavoille, David Peleg, Stéphane Pérennes, and Ran Raz. Distance labeling in graphs. In 12^{th} *Symposium on Discrete Algorithms (SODA)*. ACM-SIAM, January 2001. To appear.

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

[LB62] C. G. Lekkerkerker and J. Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fund. Math.*, 51:45–64, 1962.

[LUW95] F. Lazebnik, V. A. Ustimenko, and A. J. Woldar. A new series of dense graphs of high girth. *Bulletin of American Mathematical Society (New Series)*, 32(1):73–79, 1995.

[MR97] J. Ian Munro and Venkatesh Raman. Succinct representation of balanced parentheses, static trees and planar graphs. In IEEE, editor, 38^{th} *Symposium on Foundations of Computer Science (FOCS)*, pages 118–126, October 1997.

[Mun96] J. Ian Munro. Tables. In 16^{th} *FST&TCS*, volume 1180 of Lectures Notes in Computer Science, pages 37–42. Springer-Verlag, 1996.

[Pel99] David Peleg. Proximity-preserving labeling schemes and their applications. In Peter Widmayer, Gabriele Neyer, and Stephan Eidenbenz, editors, 25^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 1665 of Lecture Notes in Computer Science, pages 30–41. Springer, June 1999.

[RS86] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.

[Win83] Peter Winkler. Proof of the squashed cube conjecture. *Combinatorica*, 3(1):135–139, 1983.