Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Algorithms 53 (2004) 85–112
Journal of Algorithms
www.elsevier.com/locate/jalgor

Distance labeling in graphs ☆

Cyril Gavoille [a], David Peleg [b,*,1], Stéphane Pérennes [c], Ran Raz [b]

[a] *LaBRI, Université Bordeaux 1, 351, cours de la Libération, 33405 Talence Cedex, France*
[b] *Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, 76100 Israel*
[c] *SLOOP I3S-CNRS/INRIA, Université de Nice-Sophia Antipolis, 2004 Route des Lucioles, BP 93, F-06902 Sophia Antipolis Cedex, France*

Received 1 November 2002

Available online 15 June 2004

Abstract

We consider the problem of labeling the nodes of a graph in a way that will allow one to compute the distance between any two nodes directly from their labels (without using any additional information). Our main interest is in the minimal length of labels needed in different cases. We obtain upper and lower bounds for several interesting families of graphs. In particular, our main results are the following. For general graphs, we show that the length needed is $\Theta(n)$. For trees, we show that the length needed is $\Theta(\log^2 n)$. For planar graphs, we show an upper bound of $O(\sqrt{n}\log n)$ and a lower bound of $\Omega(n^{1/3})$. For bounded degree graphs, we show a lower bound of $\Omega(\sqrt{n})$.

The upper bounds for planar graphs and for trees follow by a more general upper bound for graphs with a $r(n)$-separator. The two lower bounds, however, are obtained by two different arguments that may be interesting in their own right.

We also show some lower bounds on the length of the labels, even if it is only required that distances be approximated to a multiplicative factor s. For example, we show that for general graphs the required length is $\Omega(n)$ for every $s < 3$. We also consider the problem of the time complexity of the distance function once the labels are computed. We show that there are graphs with optimal labels of length $3\log n$, such that if we use any labels with fewer than n bits per label, computing the distance function requires exponential time. A similar result is obtained for planar and bounded degree graphs.

☆ This work has been supported in part by AFIRST.

* Corresponding author.
E-mail addresses: gavoille@labri.fr (C. Gavoille), david.peleg@weizmann.ac.il (D. Peleg), stephane.perennes@sophia.inria.fr (S. Pérennes), ran.raz@weizmann.ac.il (R. Raz).

1 Supported in part by grants from the Israel Science Foundation and the Israel Ministry of Science and Art.

doi:10.1016/j.jalgor.2004.05.002

1. Introduction

1.1. Motivation

Most common network representations are global in nature, and require users to have access to data on the entire network structure in order to derive useful information, even if the sought piece of information is very local, and pertains to only few nodes.

In contrast, the notion of *adjacency labeling schemes*, introduced by Breuer and Folkman [2,3], involves using more *localized* labeling schemes for networks. The idea is to label the nodes in a way that will allow one to infer the adjacency of two nodes *directly* from their labels, without using *any* additional information sources.

Obviously, labels of unrestricted size can be used to encode any desired information. However, for such a labeling scheme to be useful, it should strive to use relatively *short* labels (say, of length poly-logarithmic in n), and yet allow efficient (say, poly-logarithmic time) information deduction. The feasibility of such *efficient* adjacency labeling schemes was explored over a decade ago by Kannan, Naor and Rudich [7].

Interest in this natural idea was recently revived by the observation that in addition to *adjacency* labeling schemes, it may be possible to devise similar schemes for capturing *distance* information. This has led to the notion of *distance labeling schemes*, which are schemes possessing the ability to determine the distance between two nodes efficiently (say, in poly-logarithmic time again) given their labels [11].

The relevance of distance labeling schemes in the context of communication networks has been pointed out in [11], and illustrated by presenting an application of such labeling schemes to distributed connection setup procedures in circuit-switched networks. It seems very plausible that distance labeling schemes may be useful also in the design of "memory-free" routing schemes, which are routing schemes geared towards supporting architectures based on very fast and simple switches, allowed to store very little data locally. Some other problems where distance labeling schemes may play an active role are bounded ("time-to-live") broadcast protocols and topology update mechanisms.

1.2. Distance labeling

Let us define the notion of distance labeling schemes more precisely. Given an undirected connected weighted graph G and two nodes u and v, we denote by $d_G(u,v)$ the distance between u and v in G, i.e., the minimum weight of a path between them. (For an unweighted graph, consider all edges to have weight 1.)

A *node-labeling* for the graph G is a function L that assigns a non-negative integer label $L(u,G)$ to each node u of G.

A *distance decoder* is a 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 $\langle L, f\rangle$ is a *distance labeling* for G if $f(L(u,G),L(v,G)) = d_G(u,v)$ for any pair of nodes $u, v \in V(G)$. More generally, $\langle L, f\rangle$ is a *distance labeling scheme* for the graph family $\mathcal{G}$ if it is a distance labeling for every graph $G \in \mathcal{G}$. Hereafter, we denote by $\mathcal{G}_n$ the sub-family containing the n-node graphs of $\mathcal{G}$.

It is important to note that the function f, responsible of the distance computation, is independent of G. Thus f can be seen as a method used to compute the distances in a decentralized fashion, given any two labels and knowing that the graph belongs to some specific family. In particular, it must be possible to define f by a constant size algorithm (depending only of the family). In contrast, the labels contain some information that can be pre-computed by considering the whole graph structure.

Clearly, a distance labeling scheme always exists for any graph family if one allows arbitrarily large labels. In this paper we are interested in 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, and denote

$$L_{\max}(G) = \max_{u \in V(G)} \bigl|L(u, G)\bigr|.$$

Given a finite graph family $\mathcal{G}$ and a distance labeling scheme $\langle L, f\rangle$ on $\mathcal{G}$, denote

$$\ell_{\langle L,f\rangle}(\mathcal{G}) = \max\bigl\{L_{\max}(G) \mid G \in \mathcal{G}\bigr\},$$
$$\ell(\mathcal{G}) = \min\bigl\{\ell_{\langle L,f\rangle}(\mathcal{G}) \mid \langle L, f\rangle \text{ is a distance labeling scheme for } \mathcal{G}\bigr\}.$$

Instead of considering the maximal label length one can prefer the *total* label length. We denote

$$\bar{\ell}_{\langle L,f\rangle}(\mathcal{G}) = \max\Biggl\{\sum_{u \in V(G)} \bigl|L(u, G)\bigr| \Bigm| G \in \mathcal{G}\Biggr\},$$
$$\bar{\ell}(\mathcal{G}) = \min\bigl\{\bar{\ell}_{\langle L,f\rangle}(\mathcal{G}) \mid \langle L, f\rangle \text{ is a distance labeling scheme for } \mathcal{G}\bigr\}.$$

We are also interested in the efficiency of the distance computation. In a *linear* distance labeling, the worst-case time complexity is proportional to the size of the inputs, i.e., to the length of the longest label.

Distance labelings can also be defined up to multiplicative stretch factor s. That is, given a distance decoder f, a node-labeling L and a real $s \geqslant 1$, we say that $\langle L, f\rangle$ is an *s-stretched distance labeling for G* if for any pair of nodes u, v of G,

$$d_G(u, v) \leqslant f\bigl(L(u, G), L(v, G)\bigr) \leqslant s \cdot d_G(u, v).$$

All the above parameters are extended to this case by adding a superscript s.

The above definitions are for the general case of weighted graphs. Below, we will work mainly with classes of unweighted graphs (unless said otherwise).

1.3. Related work

Many on-line problems on static graph collections can be solved efficiently using preprocessing and auxiliary space. However, here we insist on more localized processing, namely, answering on-line queries with local information (or labels) associated to the nodes involved in the query alone. *Adjacency* labeling schemes are studied in [7]. Specifically, it is shown how to construct $O(\log n)$-bit adjacency labeling schemes for a number of graph families, including trees, bounded arboricity graphs (including, in particular, graphs of bounded degree and graphs of bounded genus, e.g., planar graphs), various intersection-based graphs such as interval and permutation graphs, and c-decomposable graphs. It is also easy to encode the *ancestry* (or *descendance*) relation in a tree using interval-based schemes (cf. [14]).

Concerning *distance* query on general n-node graphs, Graham and Pollak proposed to label each node by a word of q_n symbols taken in $\{0, 1, *\}$ such that the distance between two nodes corresponds to the Hamming distance of the two words (the distance between $*$ and any symbol is null) [6]. Referenced as the *Squashed Cube Conjecture*, Winkler has proved that $q_n \leqslant n-1$ for every n, implying a scheme with labels of $n \log 3 \approx 1.58n$ bits,[2] although with a prohibitive $\Theta(n)$ query time to decode the distance [15].

More recently, a distance labeling scheme for weighted trees with weights from the range $[0, M)$ using $O((\log M + \log n)\log n)$ bit labels has been given in [11], and $O(\log^2 n)$ distance labeling schemes for interval and permutation graphs were presented in [8], all with $O(\log n)$ query time. The bounds for interval graphs has been later improved to $O(\log n)$ bit labels and constant query time, and extended to circular-arc graphs [4]. Queries concerning the least-common ancestor of two nodes, and related functions, can be answered with labels of length $O(\log^2 n)$ bits with $O(\log n)$ query time [10].

1.4. Our contribution

We first present some upper bounds. For the class $\mathcal{G}$ of all graphs, Winkler showed in [15] that $\ell(\mathcal{G}_n) \leqslant 1.58n$, however with a $\Theta(n)$ time to decode the distance. We show that n-node graphs can be labeled with labels of size $11n$ bits so that in time $O(\log\log n)$ the distance between two nodes can be computed given their labels only. This result is complemented by the fact that the class $\mathcal{G}$ of all n-node graphs requires labels of size $\Omega(n)$. Hence $\ell(\mathcal{G}_n) = \Theta(n)$.

We also show that classes of graphs with (recursive) $r(n)$-separators support distance labeling scheme with labels of size $O(r(n)\log^2 n)$ (the size reduces to $O(r(n)\log n)$ whenever $r(n) \geqslant n^{\varepsilon}$ for constant $\varepsilon > 0$), such that the distance can be computed in time $O(\log n)$. This general upper bound implies several results. For instance, it implies that for the family $\mathcal{P}$ of planar graphs $\ell(\mathcal{P}_n) = O(\sqrt{n}\log n)$, and for the family $\mathcal{W}$ of graphs with bounded tree-width $\ell(\mathcal{W}_n) = O(\log^2 n)$.

Our main results concern establishing some lower bounds on the size of the labels. (Some of these bounds hold even if it is only required that the distances are *approximated* to a multiplicative stretch factor s.) In particular, we prove the following bounds:

1. For the family $\mathcal{G}$ of general graphs, we prove $\ell^s(\mathcal{G}_n) \geqslant n/2 - O(1)$ and $\bar{\ell}^s(\mathcal{G}_n) \geqslant n^2/2 - O(n\log n)$, for any $s < 2$.

[2] All the logarithms are in base two.

2. For the family $\mathcal{B}^k$ of bipartite graphs whose smaller part is of size k, we prove $\bar{\ell}^s(\mathcal{B}_n^k) \geqslant k(n-k) - O(n\log n)$, for any $s < 3$, and thus that $\bar{\ell}^s(\mathcal{G}_n) \geqslant n^2/4 - O(n\log n)$, for any $s < 3$.
3. For the family $\mathcal{D}$ of graphs of maximum degree 3, we prove $\bar{\ell}(\mathcal{D}_n) = \Omega(n^{3/2})$.
4. For the family $\mathcal{P}$ of bounded degree planar graphs, we prove $\bar{\ell}(\mathcal{P}_n) = \Omega(n^{4/3})$. This answers negatively a question of [11], but leaves an intriguing gap between our upper and lower bounds.
5. For the family $\mathcal{T}$ of unweighted binary trees, we prove $\ell(\mathcal{T}_n) \geqslant \log^2 n/8 - O(\log n)$. More generally, for the family $\mathcal{T}^M$ of binary trees with integral weights from the range $[0, M)$, $M \geqslant 2$, we prove $\ell(\mathcal{T}_n^M) = \Theta((\log M + \log n)\log n)$.

Finally, we consider the problem of the time complexity of the distance function once the labels are computed. We show that there are n-node graphs with optimal labels of size $3\log n$ such that, if one uses labels with fewer than n bits, it requires an exponential time to compute the distance function. A similar result is obtained for planar graphs, and bounded degree graphs.

2. Upper bounds

2.1. General graphs

One can easily label every node of a graph with its vector of distances to all other nodes. For n-node graphs, this leads to an $O(n\log n)$ bit node-labeling with $O(1)$ time to decode the distance. On the other hand, the Squashed Cube Conjecture provides a scheme assigning shorter labels of $O(n)$ bits but with an $\Theta(n)$ time distance decoding. Using another approach, namely a family of geometrically sized dominating sets collections, we propose in this paragraph a scheme using $O(n)$ bit labels but with a $O(\log\log n)$ time distance decoding.

We start with some preliminary claims regarding dominating sets. A *ρ-dominating set* for a graph G is a set $S \subseteq V(G)$ satisfying that for every node $v \in V(G)$ there is a node $w \in S$ at distance at most ρ from it. It is well known (cf. [12]) that for every n-node connected graph G and integer $\rho \geqslant 0$, there exists a ρ-dominating set of cardinality at most $\max\{\lfloor n/(\rho+1)\rfloor, 1\}$.

A collection $\mathcal{S} = \{(S_i, \rho_i) \mid 0 \leqslant i \leqslant k\}$ such that ρ_i is a decreasing sequence of integers (with $\rho_k = 0$), S_i is a ρ_i-dominating set for G for every $0 \leqslant i \leqslant k$ and $S_k = V(G)$, is called a *dominating collection* for G. The above discussion implies the following fact, needed for later use.

Fact 2.1. *For every connected n-node graph G and $k = \lceil\log\log n\rceil$, there exists a dominating collection $\mathcal{S} = \{(S_i, \rho_i) \mid 0 \leqslant i \leqslant k\}$ for G, such that $\rho_i = 2^{k-i}$ and $|S_i| \leqslant n/2^{k-i}$ for every $i \in \{0, \ldots, k\}$.*

Let S be a ρ-dominating set for G. For every $x \in V(G)$, let $\mathrm{dom}_S(x)$ denote the *dominator* of x in S, namely, an arbitrary node $v \in S$ minimizing $d_G(x, v)$.

Lemma 2.1. *For every two nodes $x, y \in V(G)$:*

1. $d_G(\mathrm{dom}_S(x), \mathrm{dom}_S(y)) - 2\rho \leqslant d_G(x, y) \leqslant d_G(\mathrm{dom}_S(x), \mathrm{dom}_S(y)) + 2\rho$.
2. *Knowing ρ, $d_G(x, y) \bmod (4\rho + 1)$, and $d_G(\mathrm{dom}_S(x), \mathrm{dom}_S(y))$, one can compute $d_G(x, y)$.*

Proof. The first claim is immediate by the triangle inequality. The second claim follows from the observation that the first claim defines $4p + 1$ consecutive possible values for $d_G(x, y)$, exactly one of which can be congruent to $d_G(x, y)$ modulo $4p + 1$. $\square$

Our main lemma, based on a recursive construction using a dominating collection, is the following.

Lemma 2.2. *There exists a distance labeling scheme $\langle L, f\rangle$ such that for any n-node graph G, and any dominating collection $\mathcal{S} = \{(S_i, \rho_i) \mid 0 \leqslant i \leqslant k\}$ for G,*

$$L_{\max}(G) \leqslant \sum_{i=0}^{k-1} |S_{i+1}| \log(4\rho_i + 1) + |S_0| \log n + O(k \log n).$$

Moreover, f can be computed in time $O(k)$ and each label can be computed in time $O(\sum_{i=0}^{k} |S_i|)$.

Proof. Let $I = \{0, \ldots, k\}$. Recall that $\rho_{i+1} < \rho_i$ for every $0 \leqslant i \leqslant k$ (with $\rho_k = 0$), hence the sets S_i are typically progressively larger and $S_k = V(G)$. We define a sequence of functions $\{f_i\}_{i\in I}$ and of labelings $\{L^i\}_{i\in I}$ such that for $u, v \in S_i$, $f_i(L^i(u, G), L^i(v, G)) = d_G(u, v)$. The pair $\langle L^i, f_i\rangle$ is then said to be *i-valid*. We denote by $t(i)$ the maximum time needed to compute f_i, and let $a(i) = \max_{u\in S_i} |L^i(u, G)|$.

The proof is by induction. Starting with $i = 0$ we define an ordering of the nodes of S_0. The label $L^0(u, G)$ of a node u in S_0 is made of two fields:

[a] its rank order(u) in the ordering of S_0;
[b] the list $\{d_G(u, v)\}_{v\in S_0}$, given in the ordering chosen.

The distance decoder f_0 is as follows: Given two labels $L^0(x, G)$, $L^0(y, G)$, we use field [a] of $L^0(y, G)$ in order to find order(y). Then we use field [b] of $L^0(x, G)$, containing the list $\{d_G(x, v)\}_{v\in S_0}$, select the order(y)th item in this list and output this result. Clearly, the pair $\langle L^0, f_0\rangle$ is 0-valid. Also note that $a(0) = |S_0| \log n + \log n$, and the operation requires constant time,[3] i.e., $t(0) = O(1)$.

Now we proceed inductively, assuming that $\langle L^i, f_i\rangle$ is i-valid and defining L^{i+1}. For every node $u \in S_{i+1}$, we compute its dominator in S_i, $u' = \mathrm{dom}_{S_i}(u)$, and we also choose some arbitrary *ordering* of the elements of S_{i+1}. Then we assign to u a label $L^{i+1}(u, G)$ composed of the following fields:

[3] Constant meaning involving operations on $\log n$ bit words on a RAM.

[a] the label $L^i(u', G)$ assigned to u' (at most $a(i)$ bits);
[b] the rank order(u) of u in S_{i+1} ($O(\log n)$ bits);
[c] the list of values $\{d_G(u, v) \bmod (4\rho_i + 1)\}_{v \in S_{i+1}}$, given according to the ordering chosen for S_{i+1} ($|S_{i+1}| \log(4\rho_i + 1) + O(1)$ bits).

To compute $d_G(x, y)$ for $x, y \in S_{i+1}$ from the labels $L^{i+1}(x, G)$, $L^{i+1}(y, G)$ of x and y, we proceed as follows defining f_{i+1}:

1. For $x' = \mathrm{dom}_{S_i}(x)$ and $y' = \mathrm{dom}_{S_i}(y)$, obtain $L^i(x', G)$ and $L^i(y', G)$ from field [a] of $L^{i+1}(x, G)$ and $L^{i+1}(y, G)$ respectively (constant time).
2. Determine $d_G(x', y')$ by computing $f_i(L^i(x', G), L^i(y', G))$ (time $t(i)$).
3. Obtain the rank order(y) of y in S_{i+1} from field [b] of $L^{i+1}(y, G)$ (constant time).
4. Obtain $d_G(x, y) \bmod (4\rho_i + 1)$, which is the order($y$)th entry in field [c] of $L^{i+1}(x, G)$ (constant time as the list is sorted).
5. Compute $d_G(x, y)$ as in Lemma 2.1, relying on the fact that S_i is a ρ_i-dominating set (constant time).

It is easy to verify that $\langle L^{i+1}, f_{i+1}\rangle$ is $(i + 1)$-valid. Concerning the resulting label sizes and computation times, we have

$$a(i + 1) \leqslant a(i) + |S_{i+1}| \log(4\rho_i + 1) + O(\log n),$$
$$t(i + 1) \leqslant t(i) + O(1).$$

As $S_k = V(G)$, these recurrences imply the lemma. □

We now proceed with the main theorem of this section.

Theorem 2.3. *For the class $\mathcal{G}$ of general graphs, there is a distance labeling scheme $\langle L, f\rangle$ with $\ell_{\langle L,f\rangle}(\mathcal{G}_n) \leqslant 11n + O(\log n \log\log n)$. Moreover, the distance can be computed in (sub-linear) time $O(\log\log n)$ and the set of labels can be computed in time $O(n^2)$.*

Proof. The theorem is proved by first constructing a dominating collection $\mathcal{S} = \{(S_0, \rho_0), \ldots, (S_k, \rho_k)\}$, for $k = \lceil \log\log n\rceil$, as in Fact 2.1, and then applying Lemma 2.2.

Let us now calculate the size of the resulting labels. We have

$$L_{\max}(G) \leqslant \sum_{i=0}^{k-1} |S_{i+1}| \log(4\rho_i + 1) + |S_0| \log n + O(k \log n).$$

Recalling that $\rho_i = 2^{k-i}$, $|S_i| \leqslant n/2^{k-i}$ and $k = \lceil \log\log n\rceil$, we get that the second term is bounded by $(n/\log n) \cdot \log n = n$, the third term is bounded by $O(\log n \log\log n)$, and the first term is bounded by

$$\sum_{i=0}^{k-1} \frac{n}{2^{k-(i+1)}} \cdot \log\bigl(4 \cdot 2^{k-i} + 1\bigr) \leqslant n \sum_{i=0}^{k-1} \frac{k - i + 2}{2^{k-(i+1)}} \leqslant n \sum_{i=0}^{\infty} \frac{i + 3}{2^i} \leqslant 10n.$$

Hence overall, $L_{\max}(G) \leqslant 11n + O(\log n \log\log n)$.

Considering the time complexity, in order to obtain the labeling one needs to compute the dominating collection and then some dominating sets with geometric sizes. As proposed in [12], any ρ-dominating set can be constructed in $O(n)$ time once a BFS has been performed from an arbitrary node. The dominating collection can be constructed in a total of $O(n^2)$ time, and then the construction of all the labels takes $O(n^2)$ steps.

The time analysis for computing the distance $d_G(x,y)$ from the labels $L(x,G)$, $L(y,G)$ using the distance decoder $f(L(x,G),L(y,G))$, follows directly from Lemma 2.2, and the fact that here $k = \lceil \log\log n \rceil$. □

2.2. Distance labeling and separators

It is known [7] that planar graphs support a $O(\log n)$ bit adjacency labeling scheme. In contrast, we show later on (in Section 3.5) that one cannot solve the general distance labeling problem for planar graphs using labels shorter than $\Omega(n^{1/3})$ bits. Conversely, we now show that using the recursive $O(\sqrt{n})$-separator property, the problem can be solved using $O(\sqrt{n}\log n)$ bit labels.

More generally, in this section we deal with recursive $r(n)$-separators. For an n-node graph G, a subset of nodes S is a *separator* if its deletion splits G into connected components of size at most $2n/3$.

Given a class $\mathcal{G}$ of graphs and a positive non-decreasing function $r(n)$, we say that $\mathcal{G}$ has a *recursive* $r(n)$*-separator* (or simply $r(n)$*-separator*) if for every connected graph $G \in \mathcal{G}_n$ there exists a separator S of size at most $r(n)$ such that every connected component of the graph $G \setminus S$, obtained from G by removing all the nodes of S, belongs to $\mathcal{G}$. In particular this component has a separator of size at most $r(2n/3)$.

It is well known that planar graphs have an $O(\sqrt{n})$-separator. More generally, graphs of genus γ have an $O(\sqrt{\gamma n})$-separator [5], and graphs with K_k minors excluded (a complete graph with k nodes) have an $O(k\sqrt{n\log n})$-separator [13] or a $O(\sqrt{k^3 n})$-separator [1], and are conjectured to have an $O(k\sqrt{n})$-separator. Trees, series-parallel graphs, and bounded tree-width graphs, all have an $O(1)$-separator.

For a function $r(n)$, let $R(n) = \sum_{i=0}^{\log_{3/2} n} r(n(2/3)^i)$. Note that for positive non-decreasing $r(n)$, $R(n) \leqslant r(n)\log_{3/2} n$, and $R(n) = O(r(n))$ whenever $r(n) \geqslant n^{\varepsilon}$ for constant $\varepsilon > 0$. The following is a generalization of the result of [11] for trees.

Theorem 2.4. *For a family $\mathcal{G}$ of graphs with a $r(n)$-separator,*

$$\ell(\mathcal{G}_n) \leqslant O\big(R(n)\log n + \log^2 n\big).$$

Moreover the distance can be computed in $O(\log n)$ time.

Proof. Let us describe a distance labeling scheme $\langle L, f\rangle$ for the class $\mathcal{G}$. Given a graph $G \in \mathcal{G}_n$, we choose a separator S of size at most $r(n)$ for G. For any connected component A of $V(G) \setminus S$, let G_A be the graph induced by the nodes of A. Let c be the number of components. Mark each component A by a unique identifier $I(A)$ from the range of integers $0, 1, 2, \ldots, c-1$. The separator S itself is assigned the identifier $I(S) = c$. We also fix an ordering of the nodes of S.

For each component G_A, we apply the distance labeling scheme $\langle L, f\rangle$ recursively. Let $\ell(n) = \ell_{\langle L,f\rangle}(\mathcal{G}_n)$. Now we define the labels for nodes of G.

A node x belonging to a component A receives a label composed of the following fields:

[a] the list of its distances to the nodes of S, given according to their fixed ordering ($O(|S| \log n)$ bits);
[b] the identifier $I(A)$ marking the component A ($O(\log n)$ bits);
[c] the label $L(x, G_A)$ (at most $\ell(|A|)$ bits).

A node $x \in S$ is assigned a label composed of only the first two fields (where the identifier in the second field is $I(S) = c$).

For any two nodes $x, y \in V(G)$, denote their *distance via S* by

$$\hat{d}(x, y) = \min_{s\in S}\left\{d_G(x, s) + d_G(s, y)\right\}.$$

To compute the distance between x and y in G, we must consider two situations. If x and y belong to the same component A, then $d_G(x, y) = \min\{d_{G_A}(x, y) \mid \hat{d}(x, y)\}$. Otherwise, $d_G(x, y) = \hat{d}(x, y)$.

Consequently, in order to compute $d_G(x, y)$, we first compute $\hat{d}(x, y)$ using field [a] of $L(x, G)$ and $L(y, G)$. Next we compare the component identifier $I(A)$ of x and y from field [b] of $L(x, G)$ and $L(y, G)$. If they are equal and different from c then we use field [c] of $L(x, G)$ and $L(y, G)$ and get $L(x, G_A)$ and $L(y, G_A)$, which allows to compute $d_{G_A}(x, y)$. Hence we can compute $d_G(x, y)$.

Now, as $|A| \leqslant 2n/3$ and $|S| \leqslant r(n)$, it follows that the label length satisfies

$$\ell(n) \leqslant \ell(2n/3) + O\big(r(n) \log n + \log n\big),$$

solving to $\ell(n) = O(R(n) \log n) + O(\log^2 n)$. □

Corollary 2.5.

1. *For the family $\mathcal{P}$ of planar graphs, $\ell(\mathcal{P}_n) = O(\sqrt{n} \log n)$.*
2. *For the family $\mathcal{W}$ of bounded tree-width graphs, $\ell(\mathcal{W}_n) = O(\log^2 n)$.*

3. Lower bounds

Observe that for any graph G with $V(G) = \{1, \dots, n\}$ and having a distance labeling $\langle L, f\rangle$, the tuple of labels $\langle L(1, G), \dots, L(n, G)\rangle$ suffices to reconstruct G itself; we need only to test each pair of nodes in the graph to determine if they are at distance 1 (and therefore are adjacent) or more than 1. As such, for any family $\mathcal{G}$ of 2^k labeled graphs, under any labeling the tuple of labels must at least k bits long, lest two distinct graphs be assigned the same labeling. Therefore, we can conclude that

$$\bar{\ell}(\mathcal{G}_n) + O(n \log k) \geqslant k,$$

as delimiting the fields of the tuple can take at most $O(n \log k)$ bits. In particular, there exists some n-node graph G with $L_{\max}(G) \geqslant \bar{\ell}(\mathcal{G}_n)/n \geqslant n/2 - O(\log n)$, since there are $2^{\binom{n}{2}}$ labeled graphs.

In this section we present lower bounds on the maximum label length and the total label length, for the following graph classes:

(1) general graphs with small stretched distance labeling;
(2) graphs with a $r(n)$-separator and small stretched distance labeling;
(3) sparse and bounded degree graphs;
(4) planar graphs;
(5) trees.

The first four lower bounds use the same technique, which is formalized in the next subsection.

3.1. *The main lower-bound theorem*

Let $A \subseteq V_n = \{1, \ldots, n\}$, 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. Two graphs $G, H \in \mathcal{F}$ are said to *exhibit a k-gap over A* if 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 graph family $\mathcal{F}$ is an (A, k)*-family* if every two distinct graphs $G, H \in \mathcal{F}$ exhibit a k-gap over A. The family $\mathcal{F}$ is an A*-family* if there exists a real $k > 1$ such that $\mathcal{F}$ is a (A, k)-family. For such a family, we define

$$L_{\text{sum}}(A, G) = \sum_{a \in A} \bigl|L(a, G)\bigr|,$$
$$\bar{\ell}_{\langle L, f\rangle}(A, \mathcal{F}) = \max\bigl\{L_{\text{sum}}(A, G) \mid G \in \mathcal{F}\bigr\},$$
$$\bar{\ell}(A, \mathcal{F}) = \min\bigl\{\bar{\ell}_{\langle L, f\rangle}(A, \mathcal{F}) \mid \langle L, f\rangle \text{ is a distance labeling scheme for } \mathcal{F}\bigr\},$$

and similarly for s-stretched distance labeling schemes.

Theorem 3.1. *Let $\mathcal{F}$ be an (A, k)-family, for $k > 1$. Then for any stretch $s < k$:*

1. $\ell^s(\mathcal{F}) > \frac{1}{|A|} \cdot \log |\mathcal{F}| - 1$.
2. $\bar{\ell}^s(A, \mathcal{F}) > \log |\mathcal{F}| - |A| \log\log |\mathcal{F}|$.

Proof. Let $\langle L, f\rangle$ be any (s-stretched) distance labeling scheme on $\mathcal{F}$ with $s < k$. Assume that $A = \{a_1, \ldots, a_\alpha\}$. For every graph $G \in \mathcal{F}$, let $L(G) = \langle L(a_1, G), \ldots, L(a_\alpha, G)\rangle$, and let $\mathcal{L} = \{L(G) \mid G \in \mathcal{F}\}$. First, let us show that for every two distinct $G, H \in \mathcal{F}$, $L(G) \neq L(H)$, i.e., $|\mathcal{L}| = |\mathcal{F}|$.

Assume, by way of contradiction, that $L(G) = L(H)$ for some $G, H \in \mathcal{F}$, namely, $L(a_i, G) = L(a_i, H)$ for every $a_i \in A$. By definition of $\mathcal{F}$, there exists a pair $x, y \in A$ such that $d_G(x, y) \geqslant k \cdot d_H(x, y)$ or $d_H(x, y) \geqslant k \cdot d_G(x, y)$. Without loss of generality assume the former. Hence as $s < k$, we have

$$d_G(x, y) > s \cdot d_H(x, y). \tag{1}$$

Since $\langle L, f\rangle$ is s-stretched, we have

$$d_G(x, y) \leqslant f\big(L(x, G), L(y, G)\big) \quad \text{and} \quad f\big(L(x, H), L(y, H)\big) \leqslant s \cdot d_H(x, y).$$

However, since $L(a_i, G) = L(a_i, H)$ for every $a_i \in A$, we have in particular that

$$f\big(L(x, G), L(y, G)\big) = f\big(L(x, H), L(y, H)\big).$$

Hence $d_G(x, y) \leqslant s \cdot d_H(x, y)$, contradicting inequality (1).

Now we simply evaluate the cardinality of $\mathcal{L}$ according to a given restriction on the label length. Assume that $L_{\max}(G) \leqslant l$ for every $G \in \mathcal{F}$. Recalling that $|A| = \alpha$, this implies that $\mathcal{L} \subseteq \{1, \ldots, 2^{l+1} - 1\}^\alpha$ as there are $2^{l+1} - 1$ binary labels of length at most l. So, $|\mathcal{F}| = |\mathcal{L}| < 2^{(l+1)\alpha}$, and thus $l > (\log|\mathcal{F}|)/\alpha - 1$. The first claim holds considering an s-stretched distance labeling scheme $\langle L, f\rangle$ for $\mathcal{F}$ with $\ell^s(\mathcal{F}) = l$.

A slightly more complex argument implies the second claim as well. Assume $L_{\text{sum}}(A, G) \leqslant t$ for every $G \in \mathcal{F}$. As there are $\binom{\alpha+t}{\alpha}$ ways to divide a total of t bits among α nodes, this implies that

$$|\mathcal{F}| = |\mathcal{L}| \leqslant \binom{\alpha + t}{\alpha} \cdot 2^t \leqslant (\alpha + t)^\alpha \cdot 2^t. \tag{2}$$

If $\alpha + t \geqslant \log|\mathcal{F}|$ then $t \geqslant \log|\mathcal{F}| - \alpha \geqslant \log|\mathcal{F}| - \alpha \log\log|\mathcal{F}|$. If $\alpha + t < \log|\mathcal{F}|$. Then inequality (2) implies that $|\mathcal{F}| < (\log|\mathcal{F}|)^\alpha \cdot 2^t$, hence $\log|\mathcal{F}| < \alpha \log\log|\mathcal{F}| + t$. In all the cases, $t > \log|\mathcal{F}| - \alpha \log\log|\mathcal{F}|$. The second claim holds considering an s-stretched distance labeling scheme $\langle L, f\rangle$ for $\mathcal{F}$ with $\bar{\ell}^s(A, \mathcal{F}) = t$. □

Let us remark that the theorem applies, in particular, to exact (non-approximate) schemes. This requires us to interpret such a scheme over a class of n-node graphs $\mathcal{G}_n$ as an s-stretched scheme with $s = 1$, and take $k = 1 + 1/n$.

While we have shown that the total label length may be large for a subset of the nodes in a graph (the sets A discussed above), we can in fact amplify this result in the following manner: We dangle copies of a fixed graph from each of the nodes in A, and we will see that the dangled copies must have average label length roughly that of A. The proof idea is simple: If many of the dangled copies have short labels, we can actually use that labeling to create a shorter labeling for the nodes of A.

More formally, we define a (β, δ)-graph as a graph T with a node, called *root*, at distance at most $\delta > 0$ from β other nodes. Given a (β, δ)-graph T and an A-family $\mathcal{F}$, we create for each $G \in \mathcal{F}$ a graph $\Psi_T(G)$ composed of G in which one join a copy of T to each node $a \in A$ such that a and the root of the copy of T coincide. Denote by $\mathcal{F} \circ T$ the family of graphs $\{\Psi_T(G) \mid G \in \mathcal{F}\}$.

Lemma 3.2. *Let $\mathcal{F}$ be an A-family, let T be a (β, δ)-graph, and let $\langle L, f\rangle$ be any distance labeling scheme on the family $\mathcal{F} \circ T$. Then*

$$\bar{\ell}_{\langle L,f\rangle}(\mathcal{F} \circ T) \geqslant \beta \cdot \bar{\ell}(A, \mathcal{F}) - |A|\beta\lceil\log(\delta + 1)\rceil.$$

Proof. Consider a graph $\Psi_T(G) \in \mathcal{F} \circ T$. Let $A = \{a_1, \ldots, a_\alpha\}$, and for every $i \in \{1, \ldots, \alpha\}$, let $B_i = \{b^i_1, \ldots, b^i_\beta\}$ be a subset of nodes of the copy of T associated with

a_i that are at distance at most δ from the root a_i. Partition $Z=\bigcup_{i=1}^{\alpha} B^i$ into β disjoint sets $A_t=\{b_t^1,\dots,b_t^{\alpha}\}$, $1\leqslant t\leqslant\beta$, each of cardinality α.

The idea is to construct a labeling L^t for each node of A_t, based on the labeling L in $\Psi_T(G)$, so that if any A_t has small average label length, the labeling L^t can be used to shrink the labeling for A in G.

For every $1\leqslant t\leqslant\beta$, define the distance labeling scheme $\langle L^t, f^*\rangle$ on $\mathcal{F}$ as follows. For every $G\in\mathcal{F}$, and $u\in V(G)$,

$$L^t(u,G)=\begin{cases}\langle L(u,\Psi_T(G)),0\rangle, & \text{if } u\notin A,\\ \langle L(b_t^i,\Psi_T(G)),d_T(b_t,r)\rangle, & \text{if } u=a_i\in A\end{cases}$$

and $f^*(\langle\lambda_1,d_1\rangle,\langle\lambda_2,d_2\rangle)=f(\lambda_1,\lambda_2)-(d_1+d_2)$. Clearly, f^* returns the correct distance between any two nodes of $V(G)\setminus A$ (as the fields d_1 and d_2 are null, and $V(G)\subset V(\Psi_T(G))$). For a node $a_i\in A$, we note that a_i is a cut-vertex in $\Psi_T(G)$. Thus, $d_G(a_i,u)=d_{\Psi_T(G)}(b_t^i,u)-d_{\Psi_T(G)}(b_t^i,a_i)$, for every $u\in V(G)$. Moreover, $d_{\Psi_T(G)}(b_t^i,a_i)=d_T(b_t,r)$. So, f^* is a distance decoder for L^t. Note that f^* does not depend on t.

For every t and i, $|L^t(a_i,G)|\leqslant|L(b_t^i,\Psi_T(G))|+\lceil\log(\delta+1)\rceil$, because the second field of L^t labels has $\delta+1$ possible values (namely, the code 0, and $d_T(b_t,r)\in[1,\delta]$).

We now define a labeling on $\mathcal{F}$ using the labeling L^t which minimizes $\sum_{i=1}^{\alpha}|L^t(a_i,G)|$. Let L^* be the restriction of this minimal labeling to $V(G)$. Since f^* is a distance decoder for every L^t, we conclude that f^* is a distance decoder for L^*.

Since $\langle L^*, f^*\rangle$ is a distance labeling scheme on $\mathcal{F}$ there exists $G_0\in\mathcal{F}$ such that $L^*_{\mathrm{sum}}(A,G_0)\geqslant\bar{\ell}(A,\mathcal{F})$ and hence for every $1\leqslant t\leqslant\beta$, we have $L^t_{\mathrm{sum}}(A,G_0)\geqslant\bar{\ell}(A,\mathcal{F})$. Denote $H_0=\Psi_T(G_0)$. It follows that for every $1\leqslant t\leqslant\beta$,

$$\sum_{i=1}^{\alpha}\bigl(\bigl|L\bigl(b_t^i,H_0\bigr)\bigr|+\bigl\lceil\log(\delta+1)\bigr\rceil\bigr)\geqslant\bar{\ell}(A,\mathcal{F}).$$

Since $\sum_{i=1}^{\alpha}|L(b_t^i,H_0)|=L_{\mathrm{sum}}(A_t,H_0)$, the above inequality can be rewritten as

$$L_{\mathrm{sum}}(A_t,H_0)\geqslant\bar{\ell}(A,\mathcal{F})-\alpha\bigl\lceil\log(\delta+1)\bigr\rceil.$$

And since the sets A_t are disjoint,

$$L_{\mathrm{sum}}(Z,H_0)=\sum_{t=1}^{\beta}L_{\mathrm{sum}}(A_t,H_0)\geqslant\beta\cdot\bar{\ell}(A,\mathcal{F})-\alpha\beta\bigl\lceil\log(\delta+1)\bigr\rceil.$$

We complete the proof by noting that $\bar{\ell}_{\langle L,f\rangle}(\mathcal{F}\circ T)\geqslant L_{\mathrm{sum}}(Z,H_0)$. □

Hence we can hope to amplify the lower bound of Theorem 3.1 by a multiplicative factor of β. The family $\mathcal{F}\circ T$ contains graphs of size larger than n but smaller than $n+|A||V(T)|$. This remains $O(n)$ for (β,δ)-graphs T with $|V(T)|=O(n/|A|)$ and then we can also have $\beta=O(n/|A|)$. Thus, for suitable families $\mathcal{F}$, $\mathcal{G}$, and a graph T so that $\mathcal{F}\subseteq\mathcal{G}$, and $\mathcal{F}\circ T\subseteq\mathcal{G}$, we can have

$$\bar{\ell}(\mathcal{G}_n)\geqslant\Omega\bigl(n\cdot\ell(\mathcal{G}_n)\bigr)-O(n\log n).$$

Lemma 3.2 is used to prove Theorems 3.7 and 3.8.

3.2. A lower bound for general graphs

Our first application of the main lower-bound theorem is the following.

Theorem 3.3. *Let $\mathcal{G}$ be the family of general graphs, and let $s < 2$. Then*

$$\ell^s(\mathcal{G}_n) \geqslant \frac{n}{2} - O(1) \quad \textit{and} \quad \bar{\ell}^s(\mathcal{G}_n) \geqslant \frac{n^2}{2} - O(n \log n).$$

Proof. Let $\mathcal{F}$ be the family of all labeled graphs of diameter 2 on $V_n = \{1, \ldots, n\}$. $\mathcal{F}$ is a $(V_n, 2)$-family, because for any two distinct graphs G, H of $\mathcal{F}$ there always exists a pair (x, y) of V_n for which either $d_G(x, y) = 1$ and $d_H(x, y) = 2$, or $d_G(x, y) = 2$ and $d_H(x, y) = 1$.

To apply the main lower bound theorem we need to estimate $|\mathcal{F}|$. Let $\mathcal{G}$ be the set of all (connected or disconnected) graphs on V_n. Clearly $\mathcal{F} \subset \mathcal{G}$ and $|\mathcal{G}| = 2^{\binom{n}{2}}$. Let us bound the probability that a graph G taken uniformly at random from $\mathcal{G}$ is in $\mathcal{F}$. One possible way for taking a graph G randomly and uniformly from $\mathcal{G}$ consists of setting all the possible edges with probability $p = 1/2$. Note that $G \notin \mathcal{F}$ if and only if there is a pair $x, y \in V_n$ such that x and y are not adjacent, and such that there is no $z \in V_n \setminus \{x, y\}$ adjacent to both x and y. This occurs, for a given pair $\{x, y\}$, with probability $p(1 - p^2)^{n-2} = 1/2 \cdot (3/4)^{n-2}$. Hence, it occurs for at least one pair with probability of at most

$$\binom{n}{2} \cdot 1/2 \cdot (3/4)^{n-2} < 1/2$$

for every sufficiently large n. Therefore, $\log |\mathcal{F}| \geqslant n(n-1)/2 - 1$. Both claims of the theorem now follow by Theorem 3.1 (noting, for the second claim, that also $|\mathcal{F}| \leqslant 2^{n^2}$ and hence $\log \log |\mathcal{F}| \leqslant 2 \log n$). □

3.3. A lower bound for graphs with $r(n)$-separator

Let $\mathcal{B}^k$ denote the set of bipartite graphs whose smaller part is of size k. Clearly, $\mathcal{B}^k$ has a k-separator. We next bound the total label length required by distance labeling schemes for $\mathcal{B}^k_n$.

For every bipartite graph in $\mathcal{B}^k_n$, let X and Y denote the two parts of nodes, with $|X| = k$, and $|Y| = n - k$. Consider the subset of graphs $\mathcal{F} \subset \mathcal{B}^k_n$ whose diameter is bounded by 3. Note that $\mathcal{F}$ is a $(V_n, 3)$-family, because for every two distinct $G, H \in \mathcal{F}$, there exists a pair $(x, y) \in X \times Y$ such that $d_G(x, y) = 1$ and $d_H(x, y) \neq 1$ (or the reverse). Since H is of diameter 3, the fact that there is no edge between x and y necessitates $d_H(x, y) = 3$.

Lemma 3.4. *For sufficiently large n and for $2 \log n \leqslant k \leqslant n/2$, $|\mathcal{F}| \geqslant 2^{k(n-k)-1}$.*

Proof. Clearly, $|\mathcal{B}^k_n| = 2^{k(n-k)}$. Let G be a random graph in $\mathcal{B}^k_n$. This can be taken by fixing an edge between each pair $(x, y) \in X \times Y$ with probability $p = 1/2$.

To count the number of graphs in $\mathcal{F}$ (relative to $\mathcal{B}^k_n$), it suffices to calculate the probability that G is of diameter 3. Let $(x, y) \in X \times Y$. Let $\mathcal{E}$ denote the event "for every

$u, v \in X$, $d_G(u, v) = 2$". The probability that $d_G(x, y) \leqslant 3$ is at least the probability that y has a neighbor in X, say x', and that the event $\mathcal{E}$ holds (since in this case $d_G(x, x') = 2$ and $d_G(x', y) = 1$). The probability that y has a neighbor in X is $1 - (1 - p)^k$. By the union bound, the probability that every $y \in Y$ has a member in X is at least $1 - (n - k) \cdot (1 - p)^k$. The probability that given a pair $\{u, v\}$ of X, there is some node $w \in Y$ connecting u and v is $1 - (1 - p^2)^{n-k}$. Thus (by the union bound) event $\mathcal{E}$ occurs with probability of at least $1 - \binom{k}{2} \cdot (1 - p^2)^{n-k}$. Therefore, the graph G is of diameter $\leqslant 3$ with probability $\widehat{P}$ satisfying

$$\widehat{P} \geqslant 1 - (n - k) \cdot (1 - p)^k - \binom{k}{2} \cdot \left(1 - p^2\right)^{n-k}.$$

Hence, for $p = 1/2$, this probability is at least

$$\widehat{P} \geqslant 1 - (n - k) \cdot 2^{-k} - \binom{k}{2} \cdot (3/4)^{n-k}.$$

Thus for sufficiently large n and for $n/2 \geqslant k \geqslant 2\log n$, the probability $\widehat{P}$ is larger than $1/2$ and the lemma follows. □

Theorem 3.5. *Let $s < 3$. Then for every sufficiently large n and for each k such that $2\log n \leqslant k \leqslant n/2$,*

$$\bar{\ell}^s\left(\mathcal{B}_n^k\right) \geqslant k(n - k) - 2n\log n.$$

Proof. Because $\mathcal{F}$ is a $(V_n, 3)$-family, by part 2 of Theorem 3.1, no s-stretched distance labeling scheme ($s < 3$) with total label length less than $\log|\mathcal{F}| - n\log\log|\mathcal{F}| - 1$ exists for the class $\mathcal{F}$, thus also for the class $\mathcal{B}_n^k$. By Lemma 3.4, $\log|\mathcal{F}| \geqslant k(n - k) - 1$. One can upper bound $|\mathcal{F}|$ by $|\mathcal{F}| \leqslant |\mathcal{B}_n^k| = 2^{k(n-k)} \leqslant 2^{n^2/4}$, yielding $\log\log|\mathcal{F}| \leqslant 2\log n - 2$. The theorem follows. □

Since $\mathcal{B}^k$ has a k-separator, from 3.5 there are graphs with an n^ε-separator (for constant $\varepsilon \leqslant 1$) that require distance labelings with labels of size $\Omega(n^\varepsilon)$. The extremal case $k = n/2$ yields an alternative proof for the $\Omega(n)$ lower bound for general graphs, which in fact holds for larger stretch values, albeit with a slightly weaker constant in the leading term.

Corollary 3.6. *Let $\mathcal{G}$ be the family of general graphs, and let $s < 3$. Then for every sufficiently large n, $\bar{\ell}^s(\mathcal{G}_n) \geqslant n^2/4 - 2n\log n$.*

3.4. A lower bound for sparse graphs

Our next question is whether there exists a distance labeling scheme with short labels, say of length $O(n^\varepsilon)$ for constant $\varepsilon < 1$, for the class of n-node graphs with $O(n)$ edges. The following theorem answers this question negatively for every $\varepsilon < 1/2$. Let $\mathcal{D}$ be the class of graphs of maximum degree three.

Theorem 3.7. *For every n, $\bar{\ell}(\mathcal{D}_n) = \Omega(n^{3/2})$.*

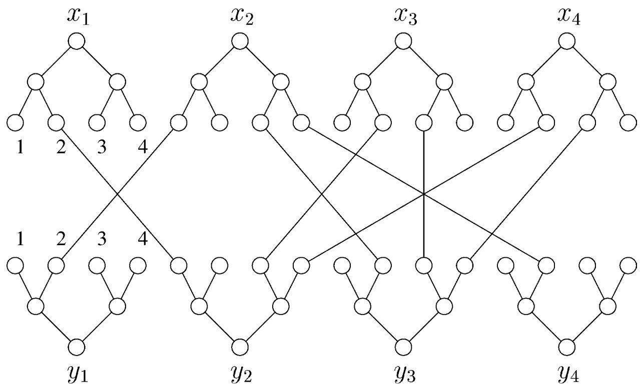

Fig. 1. A possible graph of $\mathcal{H}$ for $h = 2$.

Proof. Let $X = \{x_1, \ldots, x_{2^h}\}$ and $Y = \{y_1, \ldots, y_{2^h}\}$. We construct a family $\mathcal{H}$ of graphs defined as follows. With each node $a \in X \cup Y$ we associate a copy T_a of a complete binary tree of height h. We assume that the leaves of the trees are numbered 1 through 2^h. The union of these $2 \cdot 2^h$ trees forms the set of nodes and a part of the edge-set of all the graphs of $\mathcal{H}$. In addition, for every $(x_i, y_j) \in X \times Y$, a graph $H \in \mathcal{H}$ may or may not contain a *cross edge* $e_{i,j}$ connecting the jth leaf of T_{x_i} with the ith leaf of T_{y_j}. Thus the class $\mathcal{H}$ consists of all the graphs generated by considering all possible such choices:

$$|\mathcal{H}| = 2^{2^{2h}}.$$

Alternatively, $\mathcal{H}$ can be viewed as the class of all bipartite graphs with parts X and Y, in which every node is replaced by a complete binary tree of height h. (See Fig. 1.)

The maximum degree of each graph of $\mathcal{H}$ is three (as in particular, there is at most one cross edge touching any leaf). By the above definition, every two graphs $G, H \in \mathcal{H}$ differ on some cross edge $e_{i,j}$, and subsequently, exhibit a gap, as $d_G(x_i, y_j) \neq d_H(x_i, y_j)$. Hence $\mathcal{H}$ is an A-family. Let $A = X \cup Y$. By part 2 of Theorem 3.1, any distance labeling scheme $\langle L, f \rangle$ on $\mathcal{H}$ requires

$$\bar{\ell}(A, \mathcal{H}) \geqslant \log|\mathcal{H}| - |A| \log\log|\mathcal{H}| = 2^{2h} - 2^{h+1} \cdot 2h.$$

Consider now a complete binary tree T of height h with a node r of degree one attached to its root. The 2^h leaves of T are at distance $h+1$ from r. So T is a $(2^h, h+1)$-graph, and $\mathcal{H} \circ T \subset \mathcal{D}$. Thus, by Lemma 3.2, the family $\mathcal{H} \circ T$ satisfies

$$\begin{aligned}
\bar{\ell}(\mathcal{H} \circ T) &\geqslant 2^h \cdot \bar{\ell}(A, \mathcal{H}) - |A| \cdot 2^h \cdot \lceil \log(h+2) \rceil \\
&\geqslant 2^h \big(2^{2h} - 2^{h+1} \cdot 2h\big) - 2^{2h+1} \lceil \log(h+2) \rceil \\
&\geqslant 2^{3h} - 2^{2h+1} \big(2h + \lceil \log(h+2) \rceil\big) \\
&\geqslant 2^{3h} - O\big(h 2^{2h}\big).
\end{aligned}$$

The node set of every graph of $\mathcal{H} \circ T$ is composed of $2|A|$ trees isomorphic to a complete binary tree of height h. As each tree has $2^{h+1} - 1$ nodes, the total number of nodes of every graph in $\mathcal{H} \circ T$ is $n = 2 \cdot 2^{h+1} \cdot (2^{h+1} - 1) < 2^{2h+3}$. It follows that

$$\bar{\ell}(\mathcal{H} \circ T) \geqslant 2^{3h} - O\big(h2^{2h}\big) \geqslant (n/8)^{3/2} - O(n \log n).$$

Clearly from $\mathcal{H} \circ T$ one can form a subfamily of $\mathcal{D}$ with exactly n nodes by adding some path of suitable length to some leaf of T_a trees. This does not affect any distance of the original graph, and shows that, for every n, $\bar{\ell}(\mathcal{D}_n) = \Omega(n^{3/2})$ as claimed. □

3.5. A lower bound for planar graphs

In this subsection we provide a lower bound for planar graphs. Note that a graph with a $O(\sqrt{n}\,)$-separator is not necessarily planar. In particular, almost all the subgraphs of the complete bipartite graph $K_{\sqrt{n},n-\sqrt{n}}$ are not planar (because they contain $K_{3,3}$), and yet they have a $\sqrt{n}$-separator. So the lower bound of Theorem 3.5 cannot be applied.

Theorem 3.8. *There exists a graph family $\mathcal{P}$ consisting of bounded degree planar graphs, such that for every n, $\bar{\ell}(\mathcal{P}_n) = \Omega(n^{4/3})$.*

Proof. We first construct a class $\mathcal{G}$ of planar n-node graphs of bounded degree, which is an S-family for a node set S of size $|S| = O(n^{1/3})$, and such that $\log|\mathcal{G}| = \Omega(n^{2/3})$. Since the size of any family of n-node bounded-degree graphs is at most $2^{O(n \log n)}$, it follows by part 2 of Theorem 3.1 that every distance labeling scheme on $\mathcal{G}$ requires $\bar{\ell}(S, \mathcal{G}) \geqslant \Omega(n^{2/3})$. Then it remains to consider the family $\mathcal{P} = \mathcal{G} \circ T$, where T is a complete binary tree with $\Theta(n^{2/3})$ leaves. $\mathcal{P}$ is composed of planar $O(n)$-node bounded degree graphs, and by Lemma 3.2 every distance labeling on $\mathcal{P}$ requires $\bar{\ell}(\mathcal{G} \circ T) = \Omega(n^{4/3})$.

Description of an S-family $\mathcal{G}$

Consider the upper-left half of a grid of k columns and k rows (see Fig. 2). The node with coordinates (i, j), i.e., residing on the ith column and jth row of the grid, is named $z_{i,j}$. The set of nodes we consider in the grid is $Z = \{z_{i,j} \mid 2 \leqslant i + j \leqslant k + 1\}$ (drawn in gray in Fig. 2). At every node $z_{i,1}$, for $1 \leqslant i \leqslant k$, we attach a node u_i of degree one, and at every node $z_{k+1-j,j}$, for $1 \leqslant j \leqslant k$, we attach a node v_j of degree one. To lighten notations u_i is also named $z_{i,0}$ and v_j named $z_{k+2-j,j}$. For every $z_{i,j} \in Z$, the edge $(z_{i,j}, z_{i,j-1})$ is subdivided into two edges $(z_{i,j}, x_{i,j})$ and $(x_{i,j}, z_{i,j-1})$, adding the node $x_{i,j}$. Moreover the edge $(z_{i,j}, z_{i+1,j})$ is subdivided into the edges $(z_{i,j}, y_{i,j})$ and $(y_{i,j}, z_{i+1,j})$, adding the node $y_{i,j}$. Finally we add the edge $e_{i,j} = (x_{i,j}, y_{i,j})$ for all i, j.

We will use weights on the edges. Specifically, we assign the weight $w(e) = 1$ for every edge e, except for the edges $(x_{i,j}, z_{i,j-1})$ which are assigned the weight $2i - 1$, and the edges $(y_{i,j}, z_{i+1,j})$ which are assigned the weight $2j - 1$, for all i, j such that $2 \leqslant i + j \leqslant k + 1$. The resulting labeled graph is denoted by G_k. It is planar and of degree bounded by 4. It is depicted on Fig. 2 with $k = 6$.

It should be clear that the graph G_k can be transformed back into an unweighted graph, by replacing each edge e of weight $w(e)$ with a simple chain of $w(e)$ edges. Since an edge with weight w contributes $w - 1$ new nodes, the total number of nodes in the unweighted version of G_k is

$$n = \sum_{2 \leqslant i+j \leqslant k+1} \big(2i + 2j + O(1)\big) + O\big(k^2\big) = (2/3) \cdot k^3 + O\big(k^2\big).$$

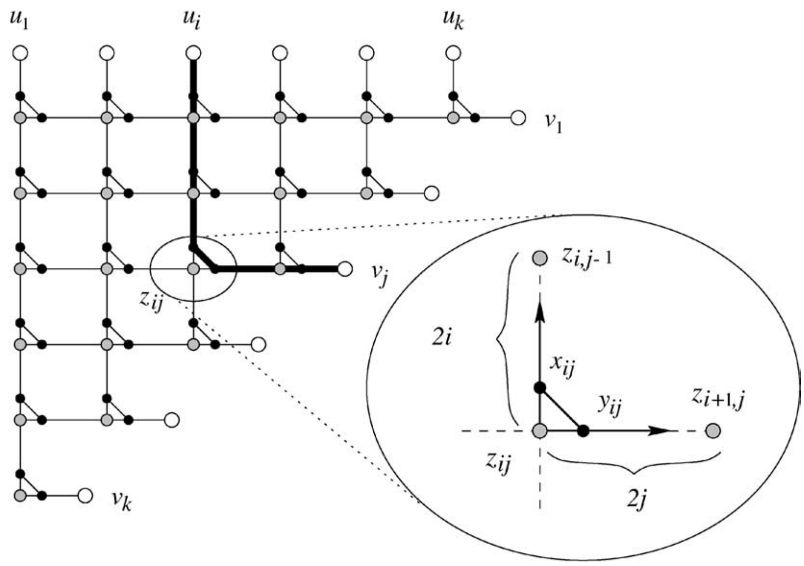

Fig. 2. The graph G_k defining $\mathcal{G}$.

For convenience, we henceforth discuss the graph in its weighted form.

Let $S = \{u_1, \ldots, u_k, v_1, \ldots, v_k\}$. The family $\mathcal{G}$ is composed of all graphs G_k in which we decide to remove or not each edge $e_{i,j}$. The number of edges $e_{i,j}$ in G_k is $|Z| = k(k+1)/2$, thus $|\mathcal{G}| = 2^{k(k+1)/2}$. We need to show that $\mathcal{G}$ is an S-family. Towards proving this, we establish the following two lemmas.

Lemma 3.9. *Any shortest path in G_k from u_i to any of the nodes $\{x_{i,j}, y_{i,j}, z_{i,j}\}$, for every j, must go through the nodes of the ith column only.*

Proof. By induction on i. The lemma holds for $i = 1$, since the weights of each edge along the first column are lower than those of corresponding edges in any other column. Now assume that the lemma holds for every $i' < i$, and let us verify that it holds for i.

Towards deriving a contradiction, assume the shortest path P starting from u_i to some node $q \in \{x_{i,j}, y_{i,j}, z_{i,j}\}$ does not follow the ith column. Note that a minor diversion, say, from some node $x_{i,j}$ to $y_{i,j}$ and back to $z_{i,j}$, does not pay. So let us first consider the case that P "strays to the right," and uses some nodes of the $(i+1)$st column. Let z_{i+1,j_0} be the first node of column $i+1$ on P, and let P' be a maximal segment of P starting at z_{i+1,j_0} and restricted entirely to columns $i' > i$. Let z_{i+1,j_1} be the last node on P'. Then the sub-path $w_{i,j_0}, z_{i+1,j_0}, P', z_{i+1,j_1}, w_{i,j_1}$ of P, where $w \in \{x, z\}$, can be replaced by the direct path P'' going from w_{i,j_0} to w_{i,j_1} on the ith column. P'' is clearly shorter than P', as P' uses vertical edges of weight at least $2(i+1)$ whereas P'' uses vertical edges of weight $2i$; contradiction.

Now assume that P "strays to the left," and uses some nodes of the $(i-1)$st column. So P departs the ith column at some node z_{i,j_0}, going through y_{i-1,j_0}, to continue on columns $i' < i$ until later returning through some node y_{i-1,j_1} to z_{i,j_1} (where again we assume that

P', the segment of P between z_{i,j_0} and z_{i,j_1}, is constrained entirely to columns to the left of i). The length of P' between z_{i,j_0} and z_{i,j_1} is $2j_0 + d_{G_k}(z_{i-1,j_0}, x_{i-1,j_1}) + 2j_1$, which by the inductive hypothesis is at least

$$(j_1 - j_0)2(i-1) - 1 + 2(j_0 + j_1) = (j_1 - j_0)2i + 4j_0 - 1.$$

However, the direct path from z_{i,j_0} to z_{i,j_1} on the ith column is of length $(j_1 - j_0)2i$, a contradiction because $j_0 \geqslant 1$. Therefore the lemma holds. □

The following lemma states that the shortest path in G_k from u_i to v_j is precisely the one highlighted in Fig. 2.

Lemma 3.10. *For every i, j, $2 \leqslant i + j \leqslant k + 1$, every shortest path in G_k from u_i to v_j goes through the sequence of nodes*

$$x_{i,1},\ z_{i,1},\ x_{i,2},\ z_{i,2},\ \ldots,\ z_{i,j-1},\ x_{i,j},\ y_{i,j},\ z_{i+1,j},\ y_{i+1,j},\ \ldots,\ z_{k+1-i,j},\ y_{k+1-i,j}.$$

Proof. Fixing i, the lemma is proved by induction on j. The claim holds for $j = 1$ because the weights on the first row are minimal.

Now assume the claim holds for every $j' < j$. Let P denote a shortest path from u_i to v_j. P cannot use any node of the i'th column with $i' < i$ because otherwise P has to go through at least one more node $z_{i,j'}$ of the ith column on its way to v_j, and then the prefix of P from u_i to $z_{i,j'}$ is not shortest, by Lemma 3.9.

So assume that P uses some nodes of the $(i+1)$st column and rows $j' > j$. In this case, we observe, by an argument similar to that of the previous proof, that the weights of the edges used by this segment of P weight at least $2(j+1)$ each, making it cheaper to continue along the jth row, leading to a contradiction.

It remains to consider the case where P uses nodes of the $(i+1)$st column and of rows $j' < j$. Assume that a part of P departs the ith column at some node y_{i,j_0}, $j_0 < j$, and reaches the jth column at some node $y_{i_0,j}$, $i_0 > i$. Without loss of generality, j_0 is maximal, and i_0 is minimal. Necessarily, P goes through the $(j-1)$st row at a node $z_{i_0,j-1}$. The length of P from y_{i,j_0} to $z_{i_1,j-1}$ is $d(y_{i,j_0}, z_{i_1,j-1}) + 2i_0$. By the inductive hypothesis, any shortest path uses nodes of the ith column and nodes of the $(j-1)$st row only. It follows that $j_0 = j$. So P follows nodes of the ith column, uses the edge $e_{i,j}$, then leaves the jth row at some node $z_{i_0,j}$, $i_0 < i_1$, to go through $z_{i_0,j-1}$ and $z_{i_1,j-1}$, and to reach finally $y_{i_1,j}$. The length of P from $z_{i_0,j}$ to $y_{i_1,j}$ is thus

$$\begin{aligned} 2i_0 + d(z_{i_0,j-1}, z_{i_1,j-1}) + 2i_1 &= 2(i_0 + i_1) + (i_1 - i_0)2(j-1) \\ &= (i_1 - i_0)2j + 4i_0. \end{aligned}$$

However, using only the jth row, the distance would be $(i_1 - i_0)2j + 1$ which is smaller than the corresponding part of P for every $i_0 \geqslant 1$, contradiction. This completes the proof of the lemma. □

It follows from Lemma 3.10 that any shortest path from u_i to v_j must use the edge $e_{i,j}$, so removing this edge from the graph increases the distance by at least 1. Moreover, this shortest path does not go through any other edge $e_{i',j'}$, showing that $d_G(u_i, v_j)$ depends only on whether $e_{i,j}$ exists or not. So, given two graphs $G, H \in \mathcal{G}$ that differs by the edge $e_{i,j}$ we have $d_G(u_i, v_j) \neq d_H(u_i, v_j)$.

Application of the lower-bound theorem

Because $\mathcal{G}$ is an S-family, we have by part 3 of Theorem 3.1 that every distance labeling scheme on $\mathcal{G}$ requires $\bar{\ell}(S, \mathcal{G}) \geqslant k(k+1)/2 - O(|S| \log k)$. We have $|S| = 2k$, $n = (2/3) \cdot k^3 + O(k^2)$, thus $k > n^{1/3} - O(n^{2/9})$, and finally

$$\bar{\ell}(S, \mathcal{G}) \geqslant \frac{1}{2} n^{2/3} - O\left(n^{4/9}\right),$$

completing the proof. □

We also trivially have a $\sqrt{n}$-lower bound for non-uniform weighted planar graphs.

Corollary 3.11. *There exists a graph family $\mathcal{P}^w$ consisting of bounded degree weighted planar graphs whose weights are non negative integers in $O(\sqrt{n})$, such that for every n, $\bar{\ell}(\mathcal{P}^w_n) = \Omega(n^{3/2})$.*

3.6. A lower bound on trees

When applying the general approach for trees, considering the set $\mathcal{F}$ of all labeled trees on the set $V_n = \{1, \dots, n\}$ as a $(V_n, 1)$-family, one gets $|\mathcal{F}| = n^{n-2}$ (known as Cayley's formula). Unfortunately, this implies only the trivial $\log n$ lower bound on the average or maximum label length.

In this section we prove a stronger lower bound, namely, that for the family $\mathcal{T}^M$ of weighted trees with integral weights from the range $[0, M)$ for $M \geqslant 2$, any distance labeling scheme requires $\ell(\mathcal{T}^M_n) = \Omega((\log M + \log n) \log n)$. This bound is tight given the $O((\log M + \log n) \log n)$ distance labeling scheme given for this class in [11]. Note that for unweighted trees we obtain a lower bound of $\Omega(\log^2 n)$.

3.6.1. The class of trees

For the lower bound proof we focus on a special class of binary weighted trees called (h, M)-*trees*, $M \geqslant 2$, defined as follows. For $h = 1$, a $(1, M)$-tree T is composed of a root with a single child and two grandchildren. An integral weight $x \in [0, M)$ is associated with each of the two edges connecting the child to the two grandchildren, and the weight $M - x$ is associated with the edge connecting the root to the child.

For $h \geqslant 2$, a (h, M)-tree is constructed by taking a $(1, M)$-tree and attaching to each of its two leaves an $(h-1, M)$-tree. Hence an (h, M)-tree contains 2^h leaves, denoted $a_1, \dots, a_{2^h}$. Let $\mathcal{C}(h, M)$ denote the class of all (h, M)-trees. Note that all of those trees have the same structure, and they differ only in their weight assignment. Figure 3 depicts a $(3, M)$-tree.

Note that a (h, 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 $(h-1, M)$-trees attached to the leaves of the top tree. The subclass of $\mathcal{C}(h, M)$ consisting of (h, M)-trees with topmost weight x is denoted $\mathcal{C}(h, M, x)$. Hence $\mathcal{C}(h, M) = \bigcup_{x=0}^{M-1} \mathcal{C}(h, M, x)$.

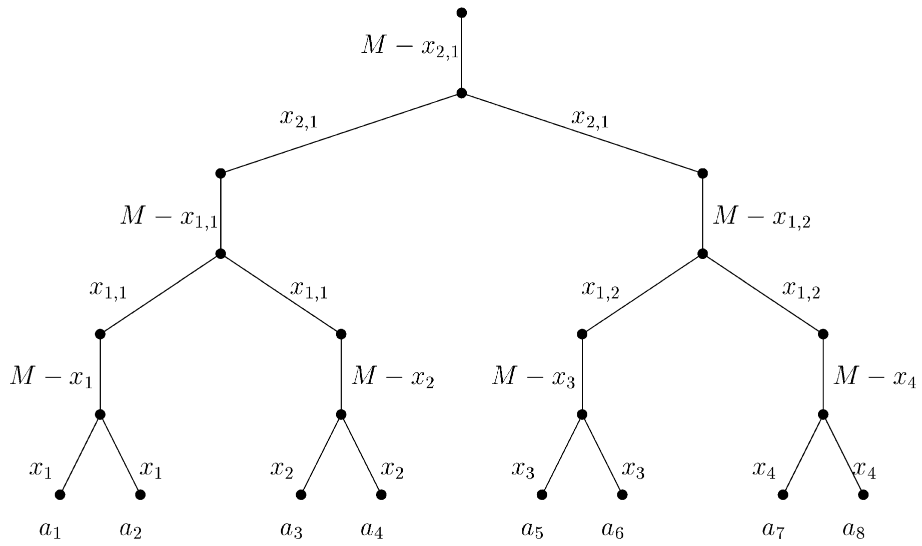

Fig. 3. A $(3, M)$-tree.

By the definition of these binary trees we have

Lemma 3.12. *For every two leaves a, a' of a tree $T \in \mathcal{C}(h, M, x)$:*

1. *If $a \in T_0$ and $a' \in T_1$ then $d_T(a, a') = 2(h-1)M + 2x$.*
2. *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 3.13. *Consider two (h, 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 $a'_1 \in T'_1$,*

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

3.6.2. The proof

For a distance labeling scheme $\langle L, f\rangle$ on $\mathcal{C}(h, M)$, let $W(L, h, M)$ denote the set of all labels assigned by L to nodes in trees of $\mathcal{C}(h, M)$, and let $g(h, M)$ denote the minimum cardinality $|W(L, h, M)|$ over all distance labeling schemes on $\mathcal{C}(h, M)$.

Hereafter, we fix $\langle L, f\rangle$ to be some distance labeling scheme attaining $g(h, M)$, i.e., such that $|W(L, h, M)| = g(h, M)$.

Let $W(x)$ denote the set of possible pairs of labels (v_0, v_1) assigned by L to some leaves $a_j \in T_0$ and $a_t \in T_1$ respectively, for some tree $T = (T_0, T_1, x) \in \mathcal{C}(h, M, x)$. Let $\mathcal{W} = \bigcup_{x=0}^{M-1} W(x)$. As $\mathcal{W} \subseteq W(L, h, M) \times W(L, h, M)$ we have

Lemma 3.14. $|\mathcal{W}| \leqslant g(h, M)^2$.

Lemma 3.15. *For every* $0 \leqslant x \neq x' < M$*, the sets* $W(x)$ *and* $W(x')$ *are disjoint.*

Proof. If there is a pair of labels in common, then the distances between each pair of nodes must be the same, which by Lemma 3.13 implies $x = x'$. □

The crux of the analysis lies in the following lemma.

Lemma 3.16. *For every* $0 \leqslant x < M$, $|W(x)| \geqslant g(h-1, M^2)$.

Proof. In any $(h-1, M^2)$-tree, a weight $w \in [0, M^2)$ can be represented by the pair of weights $w_0 = w \bmod M$, $w_1 = \lfloor w/M \rfloor$, such that $w_0, w_1 \in [0, M)$ and $w = w_0 + M w_1$.

Consequently, one can associate with any $(h-1, M^2)$-tree T' a pair of $(h-1, M)$-trees T_0 and T_1 as follows. For any edge e of T' with weight $w = w_0 + M \cdot w_1$, let the corresponding weight of e in T_0 (respectively, T_1) be w_0 (respectively, w_1). These two trees define also a (h, M)-tree $T = (T_0, T_1, x)$ in $\mathcal{C}(h, M, x)$.

Every leaf a_j of T' is now associated with two homologous leaves of T, namely, the leaf $a_j^0 = a_j$ (occurring in the left part of T, i.e., T_0), and the leaf $a_j^1 = a_{j+2^{h-1}}$ (occurring in T_1). For every two leaves a_j, a_t of T' we now have

$$d_{T'}(a_j, a_t) = d_{T_0}\big(a_j^0, a_t^0\big) + M \cdot d_{T_1}\big(a_j^1, a_t^1\big) = d_T\big(a_j^0, a_t^0\big) + M \cdot d_T\big(a_j^1, a_t^1\big). \tag{3}$$

We use this observation to derive a labeling scheme for all $(h-1, M^2)$-trees using at most $|W(x)|$ labels. Given an $(h-1, M^2)$-tree T', consider the pair of $(h-1, M)$-trees T_0, T_1 defined above, and use the labeling L to label the tree $T = (T_0, T_1, x)$. 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 $L'(a_j, T') = \langle L(a_j^0, T), L(a_j^1, T)\rangle$. Note that this pair belongs to $W(x)$.

The distance decoder f' for $(h-1, M^2)$-trees is now obtained by setting

$$\begin{aligned} f'\big(L'(a_j, T'), L'(a_t, T')\big) &= f'\big(\big\langle L\big(a_j^0, T\big), L\big(a_j^1, T\big)\big\rangle, \big\langle L\big(a_t^0, T\big), L\big(a_t^1, T\big)\big\rangle\big) \\ &= f\big(L\big(a_j^0, T\big), L\big(a_t^0, T\big)\big) + M \cdot f\big(L\big(a_j^1, T\big), L\big(a_t^1, T\big)\big). \end{aligned}$$

As L is a distance labeling scheme for (h, M)-trees we have $f(L(a_j^0, T), L(a_t^0, T)) = d_T(a_j^0, a_t^0)$ and $f(L(a_j^1, T), L(a_t^1, T)) = d_T(a_j^1, a_t^1)$, so by Eq. (3),

$$f'\big(L'(a_j, T'), L'(a_t, T')\big) = d_T\big(a_j^0, a_t^0\big) + M \cdot d_T\big(a_j^1, a_t^1\big) = d_{T'}(a_j, a_t).$$

So we have obtained a labeling scheme $\langle L', f'\rangle$ labeling any $(h-1, M^2)$-tree with labels taken from $W(x)$. It follows that $|W(x)| \geqslant g(h-1, M^2)$. □

Corollary 3.17. $g(h, M) \geqslant \sqrt{M} \cdot \sqrt{g(h-1, M^2)}$.

Proof. By Lemmas 3.15, 3.14 and 3.16, $g(h, M)^2 \geqslant |\mathcal{W}| \geqslant M \cdot g(h-1, M^2)$. □

Lemma 3.18. $g(h, M) \geqslant M^{h/2}$.

Proof. By induction on h. For $h = 1$, we have $|W(x)| \geqslant 1$, and so $|\mathcal{W}| \geqslant M$. On the other hand, by Lemma 3.14 we have $|\mathcal{W}| \leqslant g(1, M)^2$. Hence $g(1, M) \geqslant \sqrt{M}$, as claimed.

Assuming the claim for $h - 1$, we get by Corollary 3.17 and the inductive hypothesis,

$$g(h, M) \geqslant \sqrt{M}\sqrt{g(h-1, M^2)} \geqslant \sqrt{M}\sqrt{(M^2)^{(h-1)/2}} = \sqrt{M}\sqrt{M^{h-1}} = M^{h/2}. \qquad \square$$

This allows us to conclude with the lower bound.

Theorem 3.19. *For the family $\mathcal{T}^M$ of binary trees with weights from the range $[0, M)$, $M \geqslant 2$,*

$$\ell(\mathcal{T}_n^M) \geqslant \frac{1}{2}(\log n - 2)\log M.$$

Proof. By Lemma 3.18, for the class $\mathcal{C}(h, M)$ we have $\ell(\mathcal{C}(h, M)) \geqslant (h/2) \cdot \log M$. The number of nodes of the an unweighted (h, M)-tree is $n = 3 \cdot 2^h - 2$. This yields the theorem, as

$$\ell(\mathcal{T}_n^M) \geqslant \frac{1}{2} \cdot \log\left(\frac{n+2}{3}\right) \cdot \log M \geqslant \frac{1}{2}(\log n - 2)\log M. \qquad \square$$

Corollary 3.20. *For the family $\mathcal{T}$ of unweighted binary trees,*

$$\ell(\mathcal{T}_n) \geqslant \frac{1}{8}\log^2 n - O(\log n).$$

Proof. A (h, M)-tree can be transformed into an unweighted tree by replacing each edge e of weight w with a path of w (unweighted) edges. Let $t(h, M)$ be the maximal number of nodes of the unweighted tree corresponding to the construction of a (h, M)-tree. Then $t(h, M) \leqslant 3 \cdot 2^h \cdot M$. Fixing n and taking $h = \log\sqrt{n/3}$ and $M = \sqrt{n/3}$, and applying Lemma 3.18, we obtain

$$g(h, M) \geqslant 2^{\frac{h}{2}\log M} \geqslant 2^{\frac{1}{8}\log^2(n/3)}$$

for sufficiently large n. Moreover, we obtain an unweighted tree with at most $t(h, M) \leqslant n$ nodes. Note that the depth is at most $2hM < \sqrt{n}\log n$. So for unweighted binary trees with n nodes and depth $O(\sqrt{n}\log n)$, at least $\log^2 n/8 - O(\log n)$ bits may be necessary. $\square$

Corollary 3.21. *For the family $\mathcal{T}^M$ of binary trees with weights from the range $[0, M)$, $M \geqslant 2$,*

$$\ell(\mathcal{T}_n^M) = \Theta\big((\log M + \log n)\log n\big).$$

Proof. If M is bounded by some polynomial in n, we can apply Corollary 3.20 and show that $\ell(\mathcal{T}_n^M) = \Omega(\log^2 n)$. If M is super-polynomial in n, then by Theorem 3.19, $\ell(\mathcal{T}_n^M) = \Omega(\log M \log n)$. So $\ell(\mathcal{T}_n^M) = \Omega(\max\{\log^2 n, \log M \log n\}) = \Omega((\log M + \log n)\log n)$. The upper bound follows from [11]. $\square$

4. Time complexity

The time complexity for computing the distance function is sometimes crucial. For instance, a distance labeling for Cayley graphs can be built using $O(\log n)$ bit labels (by giving the full description of the generators, and labeling the nodes by a unique identifier taken from the set $\{1, \ldots, n\}$). However, there is no known efficient algorithm for computing the distance knowing the generators only. For instance, it is still an open problem to find a closed formula for the maximal distance between two nodes (namely, the diameter) for the Pancake graph, which is a Cayley graph.

In this section we show, in particular, that for every n there exists an n-node graph G for which there exists a distance labeling scheme for the class of all graphs giving it $O(\log n)$ bit labels. However, for every distance labeling scheme $\langle L, f\rangle$ that uses total label length less than $\Omega(n^2)$ for G, the time complexity of the distance function f must be larger than any (constant size) stack of exponentials. The result holds even if $\langle L, f\rangle$ has a stretch < 3.

Given a binary sequence S, let $C^{t(n)}(S|n)$ be the bounded time Kolmogorov complexity of S given n, i.e., the length of the smallest program that prints S on input n, and halts in time at most $t(n)$, where t is a *total recursive* function (namely, a function computable by a Turing Machine and defined everywhere). For every integer i, we denote S_i the ith bit of S. A *recursive* sequence is a sequence S for which there exists a Turing Machine that computes S_i for every input i. For an infinite sequence S, and every integer $n > 0$, let $[S]_n$ be the sequence composed of the first n bits of S, i.e., $[S]_n = S_1 S_2 \ldots S_n$. Using a construction of a binary sequence by diagonalization (see for instance Theorem 7.4, p. 384 of [9] for a proof), it is possible to show the following.

Lemma 4.1. *For any unbounded total recursive functions t and g there exists an infinite recursive binary sequence S such that $C^{t(n)}([S]_n|n) \geqslant n - g(n)$, for infinitely many n.*

Paraphrasing Lemma 4.1, there is a binary sequence S of length n compressible up to a constant number of bits (knowing n) for which the time to decompress any representation of S with less than $n - g(n)$ bits has arbitrarily large complexity, say $t(n)$.

For every integer $h \geqslant 0$, let $\xi_h(n)$ be the total recursive function defined by $\xi_h(n) = 2^{\xi_{h-1}(n)}$, and $\xi_0(n) = n$. This defines a stack of exponentials of fixed size h, e.g., $\xi_2(n) = 2^{2^n}$. The function ξ_h is used later for concreteness as the function t of Lemma 4.1, but it could be replace by any sufficiently large total recursive function.

Given a total recursive function τ, a family of graphs $\mathcal{F}$ is called *$\tau(n)$-recognizable* if there exists an algorithm that answers in time at most $\tau(n)$ whether $G \in \mathcal{F}$ or not for every n-node graph G. E.g., planar and bipartite graphs are respectively $O(n)$- and $O(n^2)$-recognizable families of graphs.

For every $A \subseteq V_n$, we denote by χ_A the *characteristic sequence* of A, that is the binary sequence S such that $S_i = 1$ if and only if $i \in A$.

Theorem 4.2. *Let $\mathcal{F}_n$ be a $\tau(n)$-recognizable (A, k)-family of n-node graphs such that there exists a constant h_0 such that $\xi_{h_0}(|\mathcal{F}_n|) > n$. Then, for infinitely many n, there exists $G \in \mathcal{F}_n$ with the following two properties*:

1. *There exists a distance labeling scheme* $\langle L, f\rangle$ *of stretch* 1 *on* $\mathcal{F}_n$ *such that* $L_{\max}(G) \leqslant 3\log n + O(\log\log n)$.
2. *For every distance labeling scheme* $\langle L, f\rangle$ *of stretch* $s < k$ *on* $\mathcal{F}_n$ *such that* $L_{\text{sum}}(A, G) \leqslant \log|\mathcal{F}_n| - |A|\log\log|\mathcal{F}_n| - C^{\tau(n)}(\chi_A|n) - O(\log n)$*, the space and the time complexities of* f *are greater than* $\xi_h(n) - \tau(n)$ *for any large enough constant* h.

Proof. Let $G_1, G_2, \ldots, G_i, \ldots$ be an enumeration of all graphs of $\mathcal{F}_n$; it can be defined by generating all $2^{\binom{n}{2}}$ possible n-node labeled graphs on V_n (say, in lexicographic order of their adjacency matrices) and testing whether the jth general graph belongs to $\mathcal{F}_n$ or not. Clearly, this yields also a procedure $\mathsf{Gen}(i)$ for generating the ith graph G_i in this enumeration of $\mathcal{F}_n$. Let S be a sequence satisfying Lemma 4.1 (for t, g that will be specified later on), and let TM_S be a Turing Machine computing S_i for every input i. Let $m = \lceil\log|\mathcal{F}_n|\rceil$, and let i_0 be the integer whose binary representation is the sequence $[S]_m$. We will prove the two claims for the graph $G_{i_0} \in \mathcal{F}_n$.

Short labels for G_{i_0}

Let us build a distance labeling scheme $\langle L, f\rangle$ on $\mathcal{F}_n$ for which $L_{\max}(G_{i_0}) \leqslant 3\log n + O(\log\log n)$. L is defined as follows. For every $G_i \in \mathcal{F}_n$, and for every $u \in V_n$, we set

$$L(u, G_i) = \begin{cases} \langle 0, u, i\rangle & \text{if } i \neq i_0,\\ \langle 1, u, m, \text{TM}_S\rangle & \text{if } i = i_0. \end{cases}$$

Given two labels $\lambda_1 = L(u, G)$ and $\lambda_2 = L(v, G)$, the distance between u and v in G, i.e., $f(\lambda_1, \lambda_2)$, is obtained as follows.

1. Extract the second field of λ_1 and λ_2, providing the nodes $u, v \in V_n$.
2. If the first bit of λ_1 is 0, extract from λ_1 the third field, i.
3. If the first bit of λ_1 is 1, extract m and TM_S, and compute $[S]_m = S_1 \ldots S_m$. Compute the integer i whose binary representation corresponds to the sequence $[S]_m$.
4. Invoking procedure $\mathsf{Gen}(i)$, construct the graph G_i.
5. Compute $d_{G_i}(u, v)$ with any shortest path algorithm, like Dijkstra's algorithm.

This defines a distance labeling of stretch 1 on $\mathcal{F}_n$. Let us bound $|L(u, G_{i_0})|$. The coding of an integer $z \in \{1, \ldots, Z\}$ into a binary sequence σ is said to be *self-delimiting* if z can be recovered from σ without any knowledge of Z. In particular, if σ and σ' are, respectively, two self-delimiting codings of $z \in \{1, \ldots, Z\}$ and $z' \in \{1, \ldots, Z'\}$, then the binary sequence $\sigma\sigma'$ composed of the concatenation of the bits of σ and of σ' represents a coding of the pair $\langle z, z'\rangle$. Every $z \in \{1, \ldots, Z\}$ supports a self-delimiting coding of length $\log z + O(\log\log Z)$.

To code each label $L(u, G_i)$ we use a binary string consisting of the concatenation of one bit and of the self-delimiting code of each field of the label. Therefore, for G_{i_0} we have

$$\left|L(u, G_{i_0})\right| \leqslant 1 + \log u + O(\log\log u) + \log m + O(\log\log m) + O(1),$$

noting that TM_S depends on the recursive functions t and g only and is independent of n. Thus TM_S can be coded by a constant number of bits. Because $u \in V_n$, and $m \in \{1, \ldots, \binom{n}{2}\}$, it follows that

$$L_{\max}(G_{i_0}) \leqslant 3 \log n + O(\log \log n).$$

Exponential complexity for any distance function on G_{i_0}

Consider now any distance labeling scheme $\langle L, f \rangle$ of stretch $s < k$ on $\mathcal{F}_n$. Let $t_f(n)$ be the time complexity of the distance function f.

In order to prove the second claim we will build a constant size program P that, from a suitable input of size l_P bits, and in time at most t_P, outputs $[S]_m$. As S was chosen in accordance with Lemma 4.1, it must be the case than either t_P is larger than any constant stack of exponentials, or l_P is of size larger than $m - o(m)$. Thus, expressing t_P as a function of t_f, and l_P as a function of $L_{\max}(A, \mathcal{F}_n)$, we will conclude that if $L_{\max}(A, \mathcal{F}_n)$ is too small, then t_f must be very large. More precisely, we will construct P such that l_P and t_P verifies:

1. $l_P = L_{\mathrm{sum}}(A, G_{i_0}) + |A| \log m + C^{\tau(n)}(\chi_A | n) + O(\log n)$.
2. $t_P = n^4 \cdot t_f(n) + \tau(n)^2 \cdot 2^{2n^2}$.

Assume that such a program P exists. From P and its suitable input of length l_P, one can output $[S]_m$ in time at most t_P. Thus $C^{t_P}([S]_m | m) \leqslant l_P + O(1)$. From Lemma 4.1 we conclude that: either $C^{t_P}([S]_m | m) \geqslant m - O(\log m)$ (choosing $g(m) = c \log m$ for some suitable computable constant c), or the function t_P is larger than any total recursive function in m, in particular $t_P > \xi_h(m)$ for every constant h.

Assume that $\langle L, f \rangle$ satisfies $L_{\mathrm{sum}}(A, G_{i_0}) \leqslant m - |A| \log m - C^{\tau(n)}(\chi_A | n) - O(\log n)$, i.e., the assumption of the second claim. It follows that $C^{t_P}([S]_m | m) \leqslant l_P + O(1) < m - O(\log m)$ (recall that $m < n^2$). From the above discussion, it turns out that

$$t_P = n^4 \cdot t_f(n) + \tau(n)^2 \cdot 2^{2n^2} > \xi_h(m) \quad \text{for all } m, n, \text{ and every constant } h. \tag{4}$$

Because $m \geqslant \log |\mathcal{F}_n|$ and $\xi_{h_0}(|\mathcal{F}_n|) > n$, we have for every $h > h_0$,

$$\xi_h(m) \geqslant \xi_h\bigl(\log |\mathcal{F}_n|\bigr) \geqslant \xi_{h-1}\bigl(|\mathcal{F}_n|\bigr) = \xi_{h-1-h_0}\bigl(\xi_{h_0}\bigl(|\mathcal{F}_n|\bigr)\bigr) > \xi_{h-1-h_0}(n).$$

Inequality (4) implies that $t_f(n) > \xi_h(n) - \tau(n)$ for every constant $h > h_0 + 3$. Note that any program that halts after $t_f(n)$ steps requires space at least $\log t_f(n)$.

Therefore, for such program P, the time and the space complexities of f must be larger than any stack of power of two of constant size. It remains to build such a program P.

Description of the program P

We are going to use the set of labels to determine the distances between the nodes of A, and then using procedure Gen for $\mathcal{F}$ we determine the unique graph which complies with these distances (up to the stretch factor $s < k$). This graph is G_{i_0}, and we output i_0, which is equal to $[S]_m$.

W.l.o.g. $A = \{a_1, \ldots, a_\alpha\} \subseteq V_n$ is ordered so that $a_1 < \cdots < a_\alpha$. Let $q = \binom{\alpha}{2}$. Let us consider a function μ that maps every integer $p \in \{1, \ldots, q\}$ to any pair of integers of

$\{1,\dots,\alpha\}$. Moreover assume that $\mu(p)=\{i,j\}$ can be computed in time at most $O(q)$. The choice of μ is not important, and clearly every "reasonable" $\mathbb{N}\to\mathbb{N}^2$ mapping has such time requirement (for instance the program consisting in enumerating of possible pairs of $\{1,\dots,\alpha\}$). For every $G\in\mathcal{F}_n$ and every $p\in\{1,\dots,q\}$, let $\delta_G(p)=d_G(a_i,a_j)$ where $\{i,j\}=\mu(p)$. Let $L_A=\langle L(a_1,G_{i_0}),\dots,L(a_\alpha,G_{i_0})\rangle$.

Consider the following program P on inputs $\langle L_A,\chi_A,q,n,k,f\rangle$:

1. For every $p\in\{1,\dots,q\}$, compute $\mu(p)=\{i,j\}$, and extract λ_i,λ_j respectively the ith and jth field of L_A. Compute $x_p=f(\lambda_i,\lambda_j)$.
2. For every G_i in the enumeration of $\mathcal{F}_n$, check whether $\delta_{G_i}(p)\leqslant x_p<k\cdot\delta_{G_i}(p)$ for all $p\in\{1,\dots,q\}$. If the test succeeds for all p then set $i_0=i$ and go to Step 3. This computation can be done involving $\mathsf{Gen}(i)$, and using Dijkstra's algorithm applied on the pair of nodes of A indexed by $\mu(p)$ extracted from V_n with χ_A.
3. Write i_0 in binary, yielding the sequence $[S]_m$.

Let us show that P outputs $[S]_m$. By the definition of an (A,k)-family, for every two distinct graphs $G,H\in\mathcal{F}_n$, there exists a pair of nodes of A whose distances differ by a factor at least k. Because in Step 1, $\delta_{G_{i_0}}(p)\leqslant x_p\leqslant s\cdot\delta_{G_{i_0}}(p)$ for every p, it follows that the sequences of x_p's uniquely identifies G_{i_0} in $\mathcal{F}_n$. Hence Step 2 finds i_0, and Step 3 writes $[S]_m$.

The length of the inputs is bounded by

$$L_{\text{sum}}(A,G_{i_0})+\alpha\log m+C^{\tau(n)}(\chi_A|n)+O(\log n)=l_P\quad\text{as claimed,}$$

noting that $q,n,k\in\{1,\dots,n^2\}$ (so representable on $O(\log n)$ bits), and that $L_{\text{sum}}(A,G_{i_0})+\alpha\log m$ bits are enough to describe the sequence L_A in an easy way to extract all the labels. Note also that f depends on $\mathcal{F}_n$ only (and $\mathcal{F}_n$ is $\tau(n)$-recognizable, so given n it has a constant size procedure Gen). Therefore f can be stored with a constant number of bits knowing n.

Let us compute the time complexity of P. In Step 1, there are q calls to the functions μ and f. Thus it costs $O(q(q+t_f(n)))=O(n^4+n^2t_f(n))$. In Step 2, there are 2^m calls to procedure Gen (each one costs $2^{\binom{n}{2}}\cdot\tau(n)$), and for each one there are q calls to μ, χ_A, and to Dijkstra's algorithm. Thus it costs $O(2^m\cdot2^{\binom{n}{2}}\cdot\tau(n)\cdot(n^2+\tau(n)+n^2\log n))$. In total, for n large enough and bounding $m\leqslant\binom{n}{2}$, the time is bounded by

$$O\big(n^4\cdot t_f(n)\big)+O\big(\tau(n)\cdot2^{m+\binom{n}{2}}\cdot\big(n^3+\tau(n)\big)\big)\leqslant n^4\cdot t_f(n)+\tau(n)^2\cdot2^{2n^2}=t_P$$

as required. □

For all the (A,k)-families we constructed in Section 3.1, we have $C^n(\chi_A|n)=O(1)$. Therefore, denoting by $L_{\text{sum}}(G)=\sum_{u\in V(G)}|L(u,G)|$, we have the following corollaries:

Corollary 4.3. *Let $\mathcal{G}$ be the family of all graphs. For infinitely many n there exists some graph $G\in\mathcal{G}_n$ with the following two properties*:

1. *There exists a distance labeling scheme $\langle L,f\rangle$ of stretch 1 satisfying $L_{\max}(G)\leqslant 3\log n+o(\log n)$.*

2. *For every distance labeling scheme $\langle L, f\rangle$ of stretch $s < 2$ satisfying $L_{\mathrm{sum}}(G) \leqslant n^2/2 - O(n\log n)$, the time and space complexities of the distance function f are larger than any* (*constant size*) *stack of exponentials.*

Corollary 4.4. *Let $\mathcal{B}^k$ be the family of graphs having a k-separator. For infinitely many n there exists some graph $G \in \mathcal{B}^k_n$ with the following two properties*:

1. *There exists a distance labeling scheme $\langle L, f\rangle$ of stretch 1 satisfying $L_{\max}(G) \leqslant 3\log n + o(\log n)$.*
2. *For every distance labeling scheme $\langle L, f\rangle$ of stretch $s < 3$ satisfying $L_{\mathrm{sum}}(G) \leqslant k(n-k) - O(n\log n)$, the time and space complexities of the distance function f are larger than any* (*constant size*) *stack of exponentials.*

Corollary 4.5. *Let $\mathcal{D}$ be the family of maximum degree three graphs. For infinitely many n there exists some graph $G \in \mathcal{D}_n$ with the following two properties*:

1. *There exists a distance labeling scheme $\langle L, f\rangle$ of stretch 1 satisfying $L_{\max}(G) \leqslant 3\log n + o(\log n)$.*
2. *For every distance labeling scheme $\langle L, f\rangle$ satisfying $L_{\mathrm{sum}}(G) \leqslant \Theta(n^{3/2})$, the time and space complexities of the distance function f are larger than any* (*constant size*) *stack of exponentials.*

Corollary 4.6. *Let $\mathcal{P}$ be the family of bounded degree planar graphs. For infinitely many n there exists some graph $G \in \mathcal{P}_n$ with the following two properties*:

1. *There exists a distance labeling scheme $\langle L, f\rangle$ satisfying $L_{\max}(G) \leqslant 3\log n + o(\log n)$.*
2. *For every distance labeling scheme $\langle L, f\rangle$ satisfying $L_{\mathrm{sum}}(A, G) \leqslant \Theta(n^{2/3})$ for a subset A of $O(n^{1/3})$ nodes, the time and space complexities of the distance function f are larger than any* (*constant size*) *stack of exponentials.*

4.1. Conclusion

We have proved several upper and lower bounds on the label length required to compute distances between pair of nodes in an n-node graph for different classes. This paper leaves some open questions.

- Find the smallest constant c such that there is a distance labeling scheme on arbitrary graphs with labels of length at most $cn + o(n)$ bits. The current range for c is $c \in [1/2, \log 3]$.
- Find the complexity of the maximum label length for unweighted planar graphs, the current complexity ranging between $\Omega(n^{1/3})$ and $O(\sqrt{n}\log n)$.
- Find the complexity of the maximum label length for bounded degree graphs, the current complexity ranging between $\Omega(\sqrt{n})$ and $O(n)$.

Acknowledgments

We thank the anonymous referee for many helpful suggestions.

References

[1] N. Alon, P.D. Seymour, R. Thomas, A separator theorem for graphs with an excluded minor and its applications, in: 22nd Annual ACM Symposium on Theory of Computing (STOC), 1990, pp. 293–299.
[2] M.A. Breuer, Coding the vertexes of a graph, IEEE Trans. Inform. Theory IT-12 (1966) 148–153.
[3] M.A. Breuer, J. Folkman, An unexpected result on coding the vertices of a graph, J. Math. Anal. Appl. 20 (1967) 583–600.
[4] C. Gavoille, C. Paul, Optimal distance labeling schemes for interval and circular-arc graphs, in: 11th Annual European Symposium on Algorithms (ESA), in: Lecture Notes in Comput. Sci., vol. 2832, Springer, 2003, pp. 254–265.
[5] J.R. Gilbert, J.P. Hutchinson, R.E. Tarjan, A separation theorem for graphs of bounded genus, J. Algorithms 5 (1984) 391–407.
[6] R.L. Graham, H.O. Pollak, On embedding graphs in squashed cubes, in: Lecture Notes in Math., vol. 303, 1972, pp. 99–110.
[7] S. Kannan, M. Naor, S. Rudich, Implicit representation of graphs, in: 20th Annual ACM Symposium on Theory of Computing (STOC), Chicago, IL, 1988, pp. 334–343.
[8] M. Katz, N.A. Katz, D. Peleg, Distance labeling schemes for well-separated graph classes, in: 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS), in: Lecture Notes in Comput. Sci., vol. 1770, Springer-Verlag, 2000, pp. 516–528.
[9] M. Li, P.M.B. Vitányi, An Introduction to Kolmogorov Complexity and Its Applications, Springer-Verlag, 1993.
[10] D. Peleg, Informative labeling schemes for graphs, in: 25th International Symposium on Mathematical Foundations of Computer Science (MFCS), in: Lecture Notes in Comput. Sci., vol. 1893, Springer, 2000, pp. 579–588.
[11] D. Peleg, Proximity-preserving labeling schemes, J. Graph Theory 33 (2000) 167–176.
[12] D. Peleg, E. Upfal, A trade-off between space and efficiency for routing tables, J. ACM 36 (1989) 510–530.
[13] S.A. Plotkin, S. Rao, W.D. Smith, Shallow excluded minors and improved graph decomposition, in: 5th Symposium on Discrete Algorithms (SODA), ACM–SIAM, 1994, pp. 462–470.
[14] N. Santoro, R. Khatib, Labelling and implicit routing in networks, The Computer Journal 28 (1985) 5–8.
[15] P. Winkler, Proof of the squashed cube conjecture, Combinatorica 3 (1983) 135–139.