



Shorter Labeling Schemes for Planar Graphs

Marthe Bonamy, Cyril Gavoille, Michal Pilipczuk

► **To cite this version:**

Marthe Bonamy, Cyril Gavoille, Michal Pilipczuk. Shorter Labeling Schemes for Planar Graphs. 2019. hal-02265219

HAL Id: hal-02265219

<https://hal.archives-ouvertes.fr/hal-02265219>

Submitted on 8 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Shorter Labeling Schemes for Planar Graphs*

Marthe Bonamy[†]

Cyril Gavoille[‡]

Michał Pilipczuk[§]

August 8, 2019

Abstract

An *adjacency labeling scheme* for a given class of graphs is an algorithm that for every graph G from the class, assigns bit strings (labels) to vertices of G so that for any two vertices u, v , whether u and v are adjacent can be determined by a fixed procedure that examines only their labels. It is known that planar graphs with n vertices admit a labeling scheme with labels of bit length $(2 + o(1)) \log n$. In this work we improve this bound by designing a labeling scheme with labels of bit length $(\frac{4}{3} + o(1)) \log n$.

In graph-theoretical terms, this implies an explicit construction of a graph on $n^{4/3+o(1)}$ vertices that contains all planar graphs on n vertices as induced subgraphs, improving the previous best upper bound of $n^{2+o(1)}$.

Our scheme generalizes to graphs of bounded Euler genus with the same label length up to a second-order term. All the labels of the input graph can be computed in polynomial time, while adjacency can be decided from the labels in constant time.

Keywords: planar graphs, labeling scheme, universal graphs

*The work of Michał Pilipczuk is a part of project TOTAL that have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 677651). The work of Marthe Bonamy and Cyril Gavoille is partially funded by the French ANR projects ANR-16-CE40-0023 (DESCARTES) and ANR-17-CE40-0015 (DISTANCIA).

[†]CNRS-LaBRI, University of Bordeaux, France, marthe.bonamy@u-bordeaux.fr.

[‡]LaBRI, University of Bordeaux, France, gavoille@labri.fr.

[§]Institute of Informatics, University of Warsaw, Poland, michal.pilipczuk@mimuw.edu.pl.



European Research Council
Established by the European Commission



1 Introduction

When representing graphs, say with adjacency lists or matrices, vertex identifiers usually do not play any particular role with respect to the structure of the graph: they are essentially just pointers in the data structure. In contrast, a graph is *implicitly represented* when each vertex of the graph is associated to more information so that adjacency, for instance, can be efficiently determined from the identifiers without the need of any global data-structure (cf. [KNR88, Spi03]). For instance, if G is an interval graph with n vertices, one can associate with each vertex u some interval $I(u) \subseteq [1, 2n]$ with integer endpoints so that u, v are adjacent if and only if $I(u) \cap I(v) \neq \emptyset$. Clearly, no adjacency lists or matrices are required anymore. Although G may have a quadratic number of edges, such an implicit representation uses $2 \log n + O(1)$ bits per vertex¹, regardless of its degree, which is asymptotically optimal [GP08]. Compact representations have several advantages, not only for the memory storage, but also from algorithmic perspectives. For instance, given a succinct representation, BFS traversal can be done in $O(n)$ time [RLDL94, ACJR19], even if the graph has $\Omega(n^2)$ edges. Speedups due to succinct representations are ubiquitous in the design of algorithms and data structures.

Formally introduced by Peleg [Pel00, Pel05], *informative labeling schemes* present a way to formalize implicit representations of graphs. For a given function Π defined on pairs of vertices of a graph from some given class of graphs, an informative labeling scheme has two components: an *encoding algorithm* that associates with each vertex a piece of information (label); and a *decoding algorithm* that computes $\Pi(u, v, G)$, the value of Π applied on vertices u, v of the graph G . The input of the decoding algorithm consists solely of the labels of u and of v , with no other information provided. So, finding an implicit representation of a graph G can be restated as computing an *adjacency labeling scheme* for G , that is, an informative labeling scheme where $\Pi(u, v, G)$ is TRUE if and only if u, v are adjacent in G .

In this paper we will focus on such adjacency labeling schemes (referred to as *labeling schemes* from now on), but many functions Π other than adjacency are of great interest. Among them are ancestry [FK10b] and lowest common ancestor [AGKR04, AHGL14] in rooted trees, distance labeling [GU16, GKU16, AGHP16, FGNW17] and forbidden-set distance labeling [ACGP16], compact routing [FG01, TZ01, RT15], flow [KKKP04], and many others. We refer to [GP03], and references therein, for a survey of informative labeling schemes and their applications in distributed computing, and also to [Rot16] for a survey on recent developments in labeling schemes specialized for trees.

Planar graphs. Planar graphs are perhaps the most studied class of graphs in this area, due to the wide variety of their implicit representations. To mention just a few, planar graphs are contact graphs of circles [Koe36], of 3D boxes [Tho86], of triangles [dFOdMR94], and more recently, of L-shapes [GIP18]. They also have 1-string representations [CGO10], and their incidence graphs form posets of dimension three [Sch89]. Each of these representations leads to a labelling scheme where each vertex can be encoded using a label consisting of $O(\log n)$ bits, independent of its degree.

The first explicit bound on the label length, given by Kannan et al. [KNR88], was $4 \lceil \log n \rceil$ bits. The representation in terms of dimension-3 posets due to Schnyder [Sch89] actually implies a $3 \lceil \log n \rceil$ bit labeling, and similar bounds can be derived from polynomial sized universal graphs (cf. related work below). Using the fact that planar graphs have arboricity at most three together with a labeling scheme for forests with label length $\log n + o(\log n)$, one can achieve also a similar $3 \log n + o(\log n)$ upper bound for planar graphs, where the lower-order term $o(\log n)$ directly depends on the second-order term of the

¹Throughout the paper, we denote by $\log n$ the binary logarithm of n .

bound for forests. It was a challenging question to optimize this second-order for forests. It has been successively reduced from $O(\log \log n)$ [Chu90] to $O(\log^* n)$ [AR02], and then to a constant only recently by Alstrup et al. [ADBTk17]. As explained above, this leads to an upper bound of $3 \log n + O(1)$ for planar graphs. By significantly improving the labeling scheme for bounded treewidth graphs, namely from $O(k \log n)$ [KNR88] to $\log n + O(k \log \log n)$, Gavaille and Labourel [GL07] showed that partitioning the edges of a planar graph into two bounded treewidth subgraphs, rather than into three forests, leads to a shorter representation: with labels consisting of $2 \log n + O(\log \log n)$ bits. This is currently the best known upper bound for planar graphs.

The best known results for subclasses of planar graphs are reported on Table 1.

Graph classes (with n vertices)	Upper bound (label length in bits)	References
maximum degree-2	$\log n + O(1)$	[But09, ELO08]
caterpillars	$\log n + O(1)$	[BGL06]
bounded degree trees	$\log n + O(1)$	[Chu90]
bounded depth trees	$\log n + O(1)$	[FK10a]
trees	$\log n + O(1)$	[ADBTk17]
bounded degree outerplanar	$\log n + O(1)$	[Chu90, AR14]
outerplanar	$\log n + O(\log \log n)$	[GL07]
bounded treewidth planar	$\log n + O(\log \log n)$	[GL07]
maximum degree-4 planar	$\frac{3}{2} \log n + O(\log \log n)$	[AR14]
bounded degree planar	$2 \log n + O(1)$	[Chu90]
planar	$2 \log n + O(\log \log n)$	[GL07]
diameter- d planar	$\log n + \log d + O(\log \log n)$	[this paper]
planar	$\frac{4}{3} \log n + O(\log \log n)$	[this paper]

Table 1: State-of-the-art for adjacency labeling schemes on planar graphs and some subclasses. The bounds from references [Chu90, ELO08, But09] come from induced-universal graphs, whereas all the others come from labeling schemes. The only known lower bound for planar graphs is $\log n + \Omega(1)$.

Our contribution. As shown in Table 1, in this work we present a new labeling scheme for planar graphs that uses labels of length bounded² by $\frac{4}{3} \log n$. Note that this not only improves the previously best known bound of $2 \log n$ for general planar graphs [GL07], but even the refined bound of $\frac{3}{2} \log n$ for the case of planar graphs of maximum degree 4 [AR14]. Our contribution is actually three-fold.

First, we design a labeling scheme with labels of length $\log n + \log d$ if the graph has diameter at most d (Theorem 5 in Section 4). This parametrized bound is never worse than the currently best known bound of $2 \log n$, because we always have $d < n$. Our scheme is based on a recent decomposition theorem, which states that the vertices of a planar graph can be partitioned into geodesics (shortest paths) so that contracting every geodesic to a single vertex turns the graph into a graph of constant treewidth. This was first proved by Pilipczuk and Siebertz in [PS19], and then refined by Dujmović et al. [DJM⁺19] as follows: the geodesics in the partition can be selected from any fixed BFS forest of the graph. This statement was used

²For brevity, in this informal exposition we ignore terms of lower order $o(\log n)$.

in [DJM⁺19] to prove that planar graphs have bounded queue number and in [DEJ⁺19] to prove that they have bounded nonrepetitive chromatic number, which resolved two long-standing open questions in graph theory. Thus, in this work we provide another application of the result of Dujmović et al.: a construction of shorter labeling schemes for planar graphs.

The second contribution is the main one: a labeling scheme for planar graphs that uses labels of length $\frac{4}{3} \log n$ (Theorem 6 in Section 5). To achieve this, we combine the scheme for planar graphs of low diameter with the layering technique. Essentially, we compute a decomposition of the graph into strips of depth bounded by some parameter $d \in \mathbb{N}$. Strips are separated by borders whose union is a graph on $O(n/d)$ vertices and of constant treewidth. Using the results of [GL07], for this border graph we can compute a labeling with labels of length $\log(n/d)$. On the other hand, the low-diameter result provides a scheme for the strips with labels of length at most $\log n + \log d$. At this point, superposing these two schemes gives no improvement, because vertices appearing at the borders of strips have to inherit labels from both labelings: $\log n + \log d$ from the labeling of the strips and $\log n/d$ from the labeling of the border, which sums up to $2 \log n$. However, by revisiting the scheme for graphs of bounded treewidth we are able to show that for vertices at the borders of strips, the labeling for strips can use much shorter labels: only of length $\log(n/d)$ instead of $\log n + \log d$. Hence, the combined labels of border vertices are of length at most $2 \log(n/d)$, implying that every vertex receives a label of length bounded by

$$\max \{ \log n + \log d, 2 \log(n/d) \}.$$

This expression is minimized for $d = n^{1/3}$ and then evaluates to $\frac{4}{3} \log n$, the desired bound.

The third contribution is a generalization of the previous technique to graphs of bounded Euler genus. Namely, for every fixed $g \in \mathbb{N}$, we construct a labelling scheme for graphs of Euler genus at most g that uses labels of length at most $\frac{4}{3} \log n$ (Theorem 9 in Section B).

In all our labeling schemes, given the input graph we can compute the labeling of its vertices in polynomial time, while the adjacency can be determined from the labels in constant time.

Connections with universal graphs. It has been observed in [KNR88] that the design of labeling schemes with short labels is tightly connected with the construction of small induced-universal graphs. Recall that a graph \mathcal{U} is induced-universal for a given set of graphs \mathcal{S} if every graph $G \in \mathcal{S}$ is isomorphic to some induced subgraph of \mathcal{U} . Then graphs from \mathcal{S} admit a labeling scheme with k -bit labels if and only if \mathcal{S} has an induced-universal graph \mathcal{U} with at most 2^k vertices, see [KNR88]. Thus, our new labeling scheme provides an explicit construction of an induced-universal graph for n -vertex planar graphs that has $n^{4/3+o(1)}$ vertices, improving upon the previously best known bound of $n^{2+o(1)}$, derived from [GL07].

Therefore, we proved that the minimum possible number of vertices of an induced-universal graph for n -vertex planar graphs lies between $\Omega(n)$ and $n^{4/3+o(1)}$. The search for optimum bounds on the sizes of induced-universal graphs is a well-studied topic in graph theory, see for example the recent developments for general n -vertex graphs [Alo17, AKTZ15] and for n -vertex trees [ADBTk17]. We refer readers interested in this topic to the recent survey of Alstrup et al. [AKTZ19].

Apart from induced-universal graphs, there is also an alternative definition: *edge-universal graphs*. Here, we say that \mathcal{U} is *edge-universal* for a set of graph \mathcal{S} if every graph from \mathcal{S} is a subgraph of \mathcal{U} (not necessarily induced). As far as edge-universal graphs for n -vertex planar graphs are concerned, there are much more concise constructions than in the induced setting. Babai et al. [BCE⁺82] gave a construction with $O(n^{3/2})$ edges, while if one restricts the question to n -vertex planar graphs with constant maximum

degree, then the number of edges can be reduced even to $O(n)$ [Cap02]. However, in general it is unclear how edge-universal graphs can be turned into induced-universal graphs without a significant explosion in the size, see e.g. the discussion in [Chu90].

Organization. After brief preliminaries in Section 2, we revisit and strengthen the labeling scheme for graphs of bounded treewidth of Gavaille and Labourel [GL07] in Section 3. In Section 4 we provide the scheme for planar graphs of bounded diameter, while in Section 5 we treat the case of general planar graphs. We conclude in Section 6 by stating a few open problems. The scheme for graphs of bounded genus, as well as most of the technical details of the modification of the scheme for bounded treewidth graphs, are provided in the appendix.

2 Preliminaries

We use standard graph notation. For a graph G , the vertex and edge sets of G are denoted by $V(G)$ and $E(G)$, respectively. For $A \subseteq V(G)$, we write $G[A]$ for the subgraph of G induced by A and $G - A$ for the subgraph of G induced by $V(G) \setminus A$.

Labeling schemes. The following definition formalizes the concept of labeling schemes.

Definition 1. Let \mathcal{C} be a class of graphs. An adjacency labeling scheme for \mathcal{C} is a pair $\langle \lambda, \varphi \rangle$ of functions such that, for every graph $G \in \mathcal{C}$, it holds:

- λ is the Encoder that assigns to every vertex u of G a different binary string $\lambda(u, G)$; and
- φ is the Decoder that decides adjacency from the labels taken from G . More precisely, for every pair u, v of vertices of G , $\varphi(\lambda(u, G), \lambda(v, G))$ is `TRUE` if and only if u, v are adjacent in G .

The length of the labeling scheme $\langle \lambda, \varphi \rangle$ is the function $\ell: \mathbb{N} \rightarrow \mathbb{N}$ that maps every $n \in \mathbb{N}$ to the maximum length, expressed in the number of bits, of labels assigned by the Encoder in n -vertex graphs from \mathcal{C} .

In the above definition we measure the length only in terms of the vertex count n , but we can extend the definition to incorporate auxiliary graph parameters, like diameter or treewidth, in a natural way. Whenever G is clear from the context, we write $\lambda(u)$ as a shorthand for shorthand for $\lambda(u, G)$.

When speaking about the complexity of Encoder and Decoder, we assume RAM model with machine words of bit length $O(\log n)$ and unit cost arithmetic operations.

Tree decompositions. A tree decomposition of a graph G is a pair (T, β) , where T is a tree and β maps every node x of T to its bag $\beta(x) \subseteq V(G)$ so that: for every edge uv of G there exists a node x satisfying $\{u, v\} \subseteq \beta(x)$, and for every vertex u of G , the set $\{x \in V(T) : u \in \beta(x)\}$ induces a non-empty, connected subtree of T . The width of (T, β) is $\max_{x \in V(T)} |\beta(x)| - 1$, while the treewidth of G is the minimum possible width of a tree decomposition of G .

We will also use the well-known fact that planar graphs have bounded local treewidth.

Lemma 1. Every connected planar graph of radius at most ρ has treewidth at most 3ρ .

Layered partitions of planar graphs. We now recall the results of Dujmović et al. [DJM⁺19] that we will use later on. For this, we need a few auxiliary definitions. For a graph G , a *BFS forest* of G is a spanning forest of G obtained by picking any root vertex in every connected component of G , running breadth-first search from the roots, and including all the traversed edges in the forest. Two vertices of a rooted tree are *related* if one is the ancestor (not necessarily proper) of the other. A *column* of a BFS forest F is any path in F connecting related vertices. Note that every column is a shortest path between its endpoints in G , and is contained in a shortest path from the descendant endpoint to the root of the respective tree of F .

Suppose \mathcal{P} is a partition of the vertex set of a graph G . The *quotient graph* G/\mathcal{P} has \mathcal{P} as its vertex set, and two different parts $A, B \in \mathcal{P}$ are considered adjacent in G/\mathcal{P} if and only if there exists $a \in A$ and $b \in B$ such that a and b are adjacent in G .

Theorem 2 (Theorem 8 of [DJM⁺19], with adjusted terminology). *Let G be a planar graph, and let F be any BFS forest of G . Then, one can construct in polynomial time a partition \mathcal{P} of the vertex set of G such that every part of \mathcal{P} is the vertex set of a column of F and the quotient graph G/\mathcal{P} has treewidth at most 8.*

We remark that the algorithmic statement is not stated explicitly in [DJM⁺19], but a polynomial-time algorithm can be obtained by directly following the construction in the proof.

3 Bounded Treewidth Graphs

Like the construction of [GL07] for planar graphs, our result relies on the labeling scheme developed for bounded treewidth graphs.

Theorem 3 ([GL07]). *For any fixed $k \in \mathbb{N}$, graphs of treewidth at most k admit a labeling scheme of length $\log n + O(k \log \log n)$. The Encoder runs in $O(n \log n)$ time and the Decoder runs in constant time.*

In later sections we significantly rely on the combinatorics behind the proof of Theorem 3. We will need two ingredients:

- (1) an understanding of how encoding and decoding works in the labeling scheme; and
- (2) a strengthening of the result, where we can assume that a prescribed set of at most q vertices receives shorter labels, namely of length $\log q + O(k \log \log n)$.

These two properties are formally stated as follows.

Theorem 4. *For any fixed $k \in \mathbb{N}$, the class of graphs of treewidth at most k admits a labeling scheme $\langle \lambda, \varphi \rangle$ of length $\log n + O(k \log \log n)$ with the following properties:*

- (P1) *From any label a one can extract in time $O(1)$ an identifier $\iota(a)$, so that the Decoder may be implemented as follows: given a label a , one may compute in time $O(k)$ a set $\Gamma(a)$ consisting of at most k identifiers so that $\varphi(a, b)$ is TRUE if and only if $\iota(a) \in \Gamma(b)$ or $\iota(b) \in \Gamma(a)$.*
- (P2) *If the input graph G is given together with a vertex subset Q , then the scheme can assign to the vertices of Q labels of length $\log |Q| + O(k \log \log n)$.*

The Encoder works in polynomial time while the Decoder works in constant time.

Note that contrary to Theorem 3, the Encoder of Theorem 4 does not work in near-linear time.

The proof of Theorem 4 is presented in Appendix A. In general, it largely follows the approach of Gavaille and Labourel [GL07]; in particular, their scheme achieves property (P1) without any modifications. However, to achieve property (P2) we need to replace a crucial combinatorial element of the proof with a new argument. Let us briefly sketch the main ideas.

The key idea of the scheme of Gavaille and Labourel is to work on a *bi-decomposition* of the input graph G , which is a notion roughly resembling a tree decomposition. A bi-decomposition of G is a rooted tree with nodes having at most two children. Each node is assigned its *bag*, and the bags form a partition of $V(G)$. We require that whenever uv is an edge of G , the nodes whose bags contain u and v should be related.

As proved in [GL07], an n -vertex graph G of treewidth at most k admits a bi-decomposition of depth at most $\log n$ whose bags are of size $O(k \log n)$. Let us compute an orientation of G where every vertex u has at most k outneighbors; it is known that such an orientation exists for treewidth- k graphs. By working with a chordal supergraph of G of treewidth at most k , rather than on G itself, we can assume that the *complex* K_u of u , which consists of u and all its outneighbors, forms a clique in G . Then in the computed bi-decomposition, all vertices of K_u have to lie on one root-to-leaf path. The label of u then consists of the left/right encoding of this path together with the placement of u and all its outneighbors in the bags on the path. This essentially already achieves property (P1): the set $\Gamma(\lambda(u))$ consists of the identifiers of all the outneighbors of u in the computed orientation of G .

Observe that in the approach explained above it was crucial that the computed bi-decomposition has depth at most $\log n$ and not, say, $2 \log n$, because the encoding of the root-to-leaf path that contains K_u is the dominant component in the label of a vertex u . The bi-decomposition used in [GL07] is obtained by recursively splitting the graph in half using the following balanced separator claim: every graph of treewidth at most k admits a vertex partition (A, X, B) such that $|X| = O(k \log n)$, $|A|, |B| \leq n/2$, and there is no edge between A and B . To achieve property (P2), a natural approach would be to split the graph into parts A and B using a separator X of size $O(k \log n)$ so that A and B both contain at most half of all the vertices, and both contain at most half of the remaining vertices of Q . Unfortunately, the strategy used in [GL07] seems difficult to generalize for achieving both these objectives at the same time. Therefore, we replace this element of the reasoning with a completely new argument. We allow partitions that are slightly off-balanced: by a multiplicative factor of $O\left(\frac{1}{\log n}\right)$, which is fine for achieving the overall depth $\log n + O(1)$. This relaxation allows us to use grouping based on Steinitz Lemma [Sev78, SB97] to achieve balance both in terms of the total number of vertices, and in terms of the number of vertices of Q .

4 Planar Graphs of Small Diameter

We now combine Theorem 2 with our understanding of schemes for graphs of bounded treewidth in order to give short labeling schemes for planar graphs of small diameter.

Theorem 5. *The class of connected planar graphs with n vertices admits a labeling scheme of length $\log n + \log d + O(\log \log n)$, where d is the radius of the graph. The Encoder runs in polynomial time and the Decoder in constant time.*

Moreover, if the graph is provided together with a vertex subset Q , then the Encoder may assign to the vertices of Q labels of length at most $\log |Q| + \log d + O(\log \log n)$.

PROOF. We first focus on proving the initial statement, without the additional vertex subset Q . At the end we shall argue how the refined statement can be obtained using property (P2) of Theorem 4.

Let G be the input planar graph, where G has n vertices and diameter d . Fix any vertex r of G as a root vertex and let F be a BFS tree rooted at r ; then F has depth at most d . Apply Theorem 2 to G and F ; thus we can obtain, in polynomial time, a partition \mathcal{P} of G so that every part of \mathcal{P} is the vertex set of a column in F and G/\mathcal{P} has treewidth at most 8. Denote $G' = G/\mathcal{P}$ and recall that $V(G') = \mathcal{P}$. Let $\psi: V(G) \rightarrow V(G')$ be the mapping that sends every vertex of G to the part of \mathcal{P} that contains it.

Now, apply Theorem 4 to the graph G' , obtaining a labeling scheme $\kappa(\cdot)$ defined on vertices of G' with labels of length $\log n + O(\log \log n)$, Encoder working in polynomial time and Decoder in constant time. Now, we define the labeling $\lambda(\cdot)$ of G as follows. The label $\lambda(u)$ of $u \in V(G)$ consists of:

- the label $\kappa(\psi(u))$;
- the distance from u to r , written in binary;
- a 24-bit *adjacency code*, which we define in a moment.

The first two pieces of information above are of variable length, so we add to the label a prefix of (fixed) length $2 \log \log n$ that encodes their lengths, so that they can be extracted from the label in constant time.

It remains to describe the adjacency code and how the decoding is going to be performed based on it. Recall that, by property (P1), every vertex w of G' is assigned an identifier $\iota(\kappa(w))$ so that from $\kappa(w)$ one can compute a set $\Gamma(\kappa(w))$ of at most 8 identifiers with the following property: w and w' are adjacent in G' if and only if $\iota(\kappa(w)) \in \Gamma(\kappa(w'))$ or $\iota(\kappa(w')) \in \Gamma(\kappa(w))$. By ordering identifiers lexicographically, we may assume that sets returned by $\Gamma(\cdot)$ are organized as lists. (In the original scheme of [GL07], $\Gamma(\cdot)$ sets are organized into dictionary so that membership can be tested in constant time, independently of the size of $\Gamma(\cdot)$. This refinement does not matter here since the size is bounded by 8). Observe that two vertices u and u' of G may be adjacent only if the following two assertions hold:

- $w = \psi(u)$ and $w' = \psi(u')$ are adjacent in G' ; and
- $\text{dist}(u, r)$ and $\text{dist}(u', r)$ differ by at most 1.

Hence, the adjacency code assigned to u stores, for each $i \in \{1, \dots, 8\}$ and $t \in \{-1, 0, 1\}$, whether u is adjacent to the unique vertex u' with $\psi(u') = w_i$ that satisfies $\text{dist}(u, r) - \text{dist}(u', r) = t$, where w_i is the i th vertex of $\Gamma(\kappa(\psi(u)))$. Note that there is at most one u' as above, because $\psi^{-1}(w_i)$ is a column in F .

Given the above description, the decoding can be performed as follows. Suppose we are given labels $\lambda(u)$ and $\lambda(u')$ of two vertices u, u' . From these labels we consecutively compute:

- $\text{dist}(u, r)$ and $\text{dist}(u', r)$;
- labels $\kappa(\psi(u))$ and $\kappa(\psi(u'))$;
- lists $\Gamma(\kappa(\psi(u)))$ and $\Gamma(\kappa(\psi(u')))$; and
- identifiers $\iota(\kappa(\psi(u)))$ and $\iota(\kappa(\psi(u')))$.

Next, we check whether $\iota(\kappa(\psi(u))) \in \Gamma(\kappa(\psi(u')))$ or vice versa. If this is not the case, then u and u' are not adjacent. Otherwise, we check whether $\text{dist}(u, r) - \text{dist}(u', r) \in \{-1, 0, 1\}$. Again, if this is not the case, then u and u' are not adjacent. Otherwise, whether u and u' are adjacent can be read from the adjacency code of $\kappa(u)$ or of $\kappa(u')$, depending on which identifier belongs to which list.

From the above description it is clear that the Encoder for this labeling scheme runs in polynomial time, while the Decoder runs in constant time. This concludes the proof of the initial statement, without the additional vertex subset Q . For the additional statement, we simply apply the following modification: we use property (P2) of Theorem 4 to ensure that in the labeling $\kappa(\cdot)$, the vertices of $\psi(Q)$ receive labels

of length $\log |Q| + O(\log \log n)$. Thus, in $\lambda(\cdot)$ the vertices of Q receive labels of total length at most $\log |Q| + \log d + O(\log \log n)$. \square

Remark 1. In the labeling scheme of Theorem 5, we fix a BFS tree F of depth at most d and reserve $\lceil \log d \rceil$ bits in the label of each vertex u to store the distance from u to the root of F . Observe that we may modify the scheme so that for vertices whose distance from the root is either 1 or d , this piece of information takes $O(1)$ bits. Namely, using 3 first bits we store whether the distance is 1, 2, $d-1$, d , or between 3 and $d-2$. Then actual distance is recorded using $\lceil \log d \rceil$ additional bits only when it is between 2 and $d-1$. It is easy to see that using this way of storing the distances from the root of F in the label, the Decoder can verify whether two such distances differ by at most 1, even when the Decoder does not know the value of d . We will use this optimization in the next section.

5 Planar Graphs in General

Finally, we use the layering approach in combination with Theorem 5 to give a labeling scheme for general planar graphs.

Theorem 6. *Planar graphs with n vertices admit a labeling scheme of length $\frac{4}{3} \log n + O(\log \log n)$. The Encoder runs in polynomial time and the Decoder in constant time.*

PROOF. Let $G = (V, E)$ be the input planar graph on n vertices. We may assume that G is connected, as otherwise we can make it connected by adding a fresh vertex connected to one vertex from every connected component, apply the labeling scheme for the obtained graph on $n+1$ vertices, and forget the additional vertex. Note that this operation preserves planarity.

Let

$$d = \lceil n^{1/3} \rceil .$$

W.l.o.g. we assume that $d \geq 3$ (or $n > 8$). Fix any root vertex r and partition the vertices of G according to the distance from r . Let L_i be the i -th layer composed of vertices at distance exactly $i \in \mathbb{N}$ from r , possibly with $L_i = \emptyset$ if i is too large. For $a \in \{0, \dots, d-1\}$, let

$$W_a = \bigcup_{i \in \mathbb{N}: i \equiv a \pmod{d}} L_i .$$

When speaking about sets W_a , we consider indices modulo d . Then one of the sets $W_a \cup W_{a+1}$ is small in the following sense.

Claim 1. *There exists $a \in \{0, \dots, d-1\}$ such that $|W_a \cup W_{a+1}| \leq 2n^{2/3}$.*

PROOF. Observe that $\sum_{a \in [0, d)} |W_a \cup W_{a+1}| = 2n$ because every vertex belongs to exactly two of the sets $W_a \cup W_{a+1}$. Hence, for some $a \in \{0, \dots, d-1\}$ we have $|W_a \cup W_{a+1}| \leq 2n/d \leq 2n^{2/3}$. \square

Partition the edges of G into E_1 and E_2 as follows: E_1 comprises all edges with one endpoint in W_a and the other in W_{a+1} , whereas E_2 comprises all the remaining edges. Next, define subgraphs G_1 and G_2 of G as follows:

$$G_1 = (W_a \cup W_{a+1}, E_1) \quad \text{and} \quad G_2 = (V, E_2) .$$

We first show that G_1 is a very simple and small graph.

Claim 2. *The graph G_1 has at most $2n^{2/3}$ vertices and treewidth at most 6.*

PROOF. The bound on the number of vertices of G_1 is directly implied by Claim 1. For the treewidth bound, observe that every connected component of G_1 is a subgraph of the graph $G[L_i \cup L_{i+1}]$ for some $i \in \mathbb{N}$. If one modifies G by removing all vertices of layers $L_{i+2} \cup L_{i+3} \cup \dots$ and contracting all vertices of layers $L_0 \cup \dots \cup L_{i-1}$ onto r (note that these layers induce a connected subgraph of G), then the resulting graph has radius at most 2 and contains $G[L_i \cup L_{i+1}]$ as an induced subgraph. Then the claim follows from Lemma 1. \square

Let H be a graph obtained from G_2 as follows. Add a new vertex s and make s adjacent to all vertices of W_{a+1} . Moreover, if $a \neq d - 1$, add a path of length $d - a$ connecting s with r .

Claim 3. *The graph H is connected, planar, and has at most $2n$ vertices. Moreover, all vertices of H are at distance at most d from s , where vertices of W_{a+1} are exactly at distance 1 and vertices of W_a are exactly at distance d .*

PROOF. Observe that G_2 is the disjoint union of subgraphs of G induced by the vertex subsets:

$$L_0 \cup \dots \cup L_a, \quad L_{a+1} \cup \dots \cup L_{d+a}, \quad L_{d+a+1} \cup \dots \cup L_{2d+a}, \quad \dots \quad ;$$

denote these induced subgraphs by R_0, R_1, R_2, \dots in the order as above. Observe that if, for some $i \in \mathbb{N}$ in G , we remove all vertices from layers L_j for $j > (i + 1)d + a$ and contract all vertices from layers L_j for $j \leq id + a$ onto r (note that these layers induce a connected subgraph of G), then we obtain a connected planar graph R'_i which contains R_i as an induced subgraph. In fact, we have $R_0 = R'_0$ (then no contraction takes place) and $R_i = R'_i - r$ for $i \geq 1$. Note that in R'_i for $i \geq 1$, all vertices of W_{a+1} are at distance exactly 1 from r and all vertices of W_a are at distance exactly d from r . Similarly, in R'_0 all vertices of W_a are at distance exactly a from r , and there are no vertices of W_{a+1} unless $a = d - 1$, in which case the only vertex of W_{a+1} in R'_0 is r itself.

Now observe that H can be constructed from graphs R'_0, R'_1, R'_2, \dots as follows: take the disjoint union of those graphs, identify all copies of the vertex r from R'_1, R'_2, \dots into one vertex s , and connect s to the copy of r in R'_0 via a path of length $d - a - 1$ (in case $a = d - 1$, just identify this copy of r with s as well). It is easy to see that since graphs R'_0, R'_1, R'_2, \dots are planar and connected, H is also planar and connected. The assertions about the distances of vertices of W_a and W_{a+1} from s follow directly from the discussion of distances in the previous paragraph.

The fact that H has at most $2n$ vertices follows immediately from the inequality $d \leq n$. \square

We can now use Claims 2 and 3 to give the promised labeling scheme. First, by Claim 2 and Theorem 3, for the graph G_1 we may compute a labeling λ_1 with labels of length at most $\frac{2}{3} \log n + O(\log \log n)$. Second, by Claim 3 and Theorem 5, for G_2 we may compute a labeling λ_2 with labels of length at most $\log n + \log d + O(\log \log n) = \frac{4}{3} \log n + O(\log \log n)$; this is obtained by computing such a labeling for H and restricting it to G_2 . Moreover, we may construct this labeling so that all vertices of $W_a \cup W_{a+1}$ receive shorter labels, namely of length at most $\frac{2}{3} \log n + O(\log \log n)$. Here, we use Remark 1 in order to reduce the $\log d$ summand to $O(1)$, and we use $Q = W_a \cup W_{a+1}$ as the prescribed set of at most $2n^{2/3}$ vertices in order to reduce the $\log n$ summand to $\frac{2}{3} \log n + O(1)$.

Now, the label $\lambda(u)$ of any vertex u of G is as follows:

- If $u \notin W_a \cup W_{a+1}$, then $\lambda(u) = \lambda_2(u)$.
- If $u \in W_a \cup W_{a+1}$, then $\lambda(u)$ is the concatenation of $\lambda_1(u)$ and $\lambda_2(u)$.

In the second case, in order to be able to decode $\lambda_1(u)$ and $\lambda_2(u)$ from $\lambda(u)$, we append $\log \log n$ bits that indicate the length of $\lambda_1(u)$. Also, we add one bit indicating the case into which the vertex u falls.

Thus, in the first case $\lambda(u)$ is of length $\frac{4}{3} \log n + O(\log \log n)$, while in the second it is of length

$$\frac{2}{3} \log n + O(\log \log n) + \frac{2}{3} \log n + O(\log \log n) + \log \log n = \frac{4}{3} \log n + O(\log \log n).$$

Hence, the length of the constructed labeling scheme is as claimed. For the implementation of the Decoder, given labels of two vertices u and v , we simply read labels of u and v in λ_2 and λ_1 (if applicable) and check whether they are adjacent either in G_1 or in G_2 . This concludes the construction of the labeling scheme.

The above construction can be directly translated to an implementation of the Encoder in polynomial time and the Decoder in constant time. In case of Claim 1, note that an index a satisfying the claim can be found in polynomial time by checking the indices between 0 and $d - 1$ one by one. \square

6 Conclusion and Open Problems

We have proved an upper bound of $(\frac{4}{3} + o(1)) \log n$ for the length of labeling schemes for planar graphs, and more generally for graphs embeddable in a fixed surface. It seems that our approach can be reformulated so that in fact it mostly relies on the key property observed in [DJM⁺19]: graphs embeddable in a fixed surface admit an H -partition of bounded layered width, for some graph H of bounded treewidth. This together with minor-closeness and bounded local treewidth, are essential conditions under which our technique can be employed.

As proved in [DJM⁺19], the key property of admitting an H -partition of bounded layered width for a bounded treewidth graph H is enjoyed even in larger generality: by every class of graphs that are almost-embeddable (without apices) in a fixed surface. Almost-embeddable graphs play a crucial role in the Structure Theorem for proper minor-closed classes of Robertson and Seymour [RS04], as they are atomic parts of the decomposition. Thus, extending our work to all proper minor-closed classes of graphs is an interesting, and seemingly feasible research direction. In the approach of Dujmović et al. [DEJ⁺19] for the queue number, almost-embeddable graphs are treated using layered partitions, then apices attached to those graphs are managed separately, and finally the obtained queue embeddings are glued along the tree decomposition provided by the Structure Theorem. In our case, extension to almost-embeddable graphs seems to be a matter of working out technical details, adding apices is not an issue, but the last step seems the most problematic: we do not know how to combine labeling schemes along tree decompositions. Nonetheless, we believe that an extension to all proper minor-closed graph classes is possible.

The most glaring question left open by this paper is whether planar graphs admit a labeling scheme with label length $c \log n$ with some $c < 4/3$. Now that the threshold of 2 has been broken, we do not stop to optimize the second-order terms, and conjecture that $c = 1$ should be asymptotically the right answer.

Conjecture. Planar graphs admit a labeling scheme with label length $\log n + o(\log n)$.

As already pointed by [GL07], no hereditary class with no more than $n!2^{O(n)}$ labeled graphs (including trees, planar, bounded genus, bounded treewidth, minor-free, and many others) is known to require labels

of $\log n + \omega(1)$ bits. While it may well be that the trivial lower bound is in fact tight, we do not believe that planar graphs admit a labeling scheme with label length $\log n + O(1)$. Therefore, proving a lower bound greater than the trivial $\log n + \Omega(1)$ would be very interesting.

Acknowledgements. A part of this research was completed at the 7th Annual Workshop on Geometry and Graphs held at Bellairs Research Institute in March 2019.

References

- [ACGP16] I. ABRAHAM, S. CHECHIK, C. GAVOILLE, AND D. PELEG, *Forbidden-set distance labels for graphs of bounded doubling dimension*, ACM Transactions on Algorithms, 12 (2016), p. Article No. 22. doi: [10.1145/2818694](https://doi.org/10.1145/2818694).
- [ACJR19] H. ACAN, S. CHAKRABORTY, S. JO, AND S. S. RAO, *Succinct data structures for families of interval graphs*, Tech. Rep. [1902.09228v1 \[cs.DS\]](https://arxiv.org/abs/1902.09228v1), arXiv, February 2019.
- [ADBTK17] S. ALSTRUP, S. DAHLGAARD, AND M. BÆK TEJS KNUDSEN, *Optimal induced universal graphs and adjacency labeling for trees*, Journal of the ACM, 64 (2017), p. Article No. 27. doi: [10.1145/3088513](https://doi.org/10.1145/3088513).
- [AGHP16] S. ALSTRUP, C. GAVOILLE, E. B. HALVORSEN, AND H. PETERSEN, *Simpler, faster and shorter labels for distances in graphs*, in 27th Symposium on Discrete Algorithms (SODA), ACM-SIAM, January 2016, pp. 338–350. doi: [10.1137/1.9781611974331.ch25](https://doi.org/10.1137/1.9781611974331.ch25).
- [AGKR04] S. ALSTRUP, C. GAVOILLE, H. KAPLAN, AND T. RAUHE, *Nearest common ancestors: A survey and a new algorithm for a distributed environment*, Theory of Computing Systems, 37 (2004), pp. 441–456. doi: [10.1007/s00224-004-1155-5](https://doi.org/10.1007/s00224-004-1155-5).
- [AHGL14] S. ALSTRUP, E. B. HALVORSEN, AND K. GREEN LARSEN, *Near-optimal labeling schemes for nearest common ancestors*, in 25th Symposium on Discrete Algorithms (SODA), ACM-SIAM, January 2014, pp. 972–982. doi: [10.1137/1.9781611973402.72](https://doi.org/10.1137/1.9781611973402.72).
- [AKTZ15] S. ALSTRUP, H. KAPLAN, M. THORUP, AND U. ZWICK, *Adjacency labeling schemes and induced-universal graphs*, in 47th Annual ACM Symposium on Theory of Computing (STOC), ACM Press, June 2015, pp. 625–634. doi: [10.1145/2746539.2746545](https://doi.org/10.1145/2746539.2746545).
- [AKTZ19] S. ALSTRUP, H. KAPLAN, M. THORUP, AND U. ZWICK, *Adjacency labeling schemes and induced-universal graphs*, SIAM Journal on Discrete Mathematics, 33 (2019), pp. 116–137. doi: [10.1137/16M1105967](https://doi.org/10.1137/16M1105967).
- [Alo17] N. ALON, *Asymptotically optimal induced universal graphs*, Geometric and Functional Analysis, 27 (2017), pp. 1–32. doi: [10.1007/s00039-017-0396-9](https://doi.org/10.1007/s00039-017-0396-9).
- [AR02] S. ALSTRUP AND T. RAUHE, *Small induced-universal graphs and compact implicit graph representations*, in 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, November 2002, pp. 53–62. doi: [10.1109/SFCS.2002.1181882](https://doi.org/10.1109/SFCS.2002.1181882).

- [AR14] D. ADJIASHVILI AND N. ROTBART, *Labeling schemes for bounded degree graphs*, in 41st International Colloquium on Automata, Languages and Programming (ICALP), vol. 8573 of Lecture Notes in Computer Science (ARCoSS), Springer, July 2014, pp. 375–386. doi: [10.1007/978-3-662-43951-7_32](https://doi.org/10.1007/978-3-662-43951-7_32).
- [BCE⁺82] L. BABAI, F. R. K. CHUNG, P. ERDŐS, R. L. GRAHAM, AND J. H. SPENCER, *On graphs which contain all sparse graphs*, *Annals of Discrete Mathematics*, 12 (1982), pp. 21–26. doi: [10.1016/S0304-0208\(08\)73486-8](https://doi.org/10.1016/S0304-0208(08)73486-8).
- [BGL06] N. BONICHON, C. GAVOILLE, AND A. LABOUREL, *Short labels by traversal and jumping*, in 13th International Colloquium on Structural Information & Communication Complexity (SIROCCO), vol. 4056 of Lecture Notes in Computer Science, Springer, July 2006, pp. 143–156. doi: [10.1007/11780823_12](https://doi.org/10.1007/11780823_12).
- [Bod96] H. L. BODLAENDER, *A linear time algorithm for finding tree-decompositions of small treewidth*, *SIAM Journal on Computing*, 25 (1996), pp. 1305–1317. doi: [10.1137/S0097539793251219](https://doi.org/10.1137/S0097539793251219).
- [But09] S. BUTLER, *Induced-universal graphs for graphs with bounded maximum degree*, *Graphs and Combinatorics*, 25 (2009), pp. 461–468. doi: [10.1007/s00373-009-0860-x](https://doi.org/10.1007/s00373-009-0860-x).
- [Cap02] M. R. CAPALBO, *Small universal graphs for bounded-degree planar graphs*, *Combinatorica*, 22 (2002), pp. 345–359. doi: [10.1007/s004930200017](https://doi.org/10.1007/s004930200017).
- [CFK⁺15] M. CYGAN, F. V. FOMIN, Ł. KOWALIK, D. LOKSHTANOV, D. MARX, M. PILIPCZUK, M. PILIPCZUK, AND S. SAURABH, *Parameterized Algorithms*, Springer, 2015. doi: [10.1007/978-3-319-21275-3](https://doi.org/10.1007/978-3-319-21275-3).
- [CGO10] J. CHALOPIN, D. GONÇALVES, AND P. OCHEM, *Planar graphs have 1-string representations*, *Discrete & Computational Geometry*, 43 (2010), pp. 626–647. doi: [10.1007/s00454-009-9196-9](https://doi.org/10.1007/s00454-009-9196-9).
- [Chu90] F. R. K. CHUNG, *Universal graphs and induced-universal graphs*, *Journal of Graph Theory*, 14 (1990), pp. 443–454. doi: [10.1002/jgt.3190140408](https://doi.org/10.1002/jgt.3190140408).
- [DEJ⁺19] V. DUJMOVIĆ, L. ESPERET, G. JORET, B. WALCZAK, AND D. R. WOOD, *Planar graphs have bounded nonrepetitive chromatic number*, Tech. Rep. [1904.05269v1](https://arxiv.org/abs/1904.05269v1) [[math.CO](https://arxiv.org/abs/1904.05269v1)], arXiv, 2019.
- [dFOdMR94] H. DE FRAYSSEIX, P. OSSONA DE MENDEZ, AND P. ROSENSTIEHL, *On triangle contact graphs*, *Combinatorics, Probability & Computing*, 3 (1994), pp. 233–246. doi: [10.1017/S0963548300001139](https://doi.org/10.1017/S0963548300001139).
- [DJM⁺19] V. DUJMOVIĆ, G. JORET, P. MICEK, P. MORIN, T. UECKERDT, AND D. R. WOOD, *Planar graphs have bounded queue-number*, Tech. Rep. [1904.04791v2](https://arxiv.org/abs/1904.04791v2) [[cs.DM](https://arxiv.org/abs/1904.04791v2)], arXiv, May 2019.
- [ELO08] L. ESPERET, A. LABOUREL, AND P. OCHEM, *On induced-universal graphs for the class of bounded-degree graphs*, *Information Processing Letters*, 108 (2008), pp. 255–260. doi: [10.1016/j.ipl.2008.04.020](https://doi.org/10.1016/j.ipl.2008.04.020).
- [Epp00] D. EPPSTEIN, *Diameter and treewidth in minor-closed graph families*, *Algorithmica*, 27 (2000), pp. 275–291. doi: [10.1007/s004530010020](https://doi.org/10.1007/s004530010020).

- [EW18] F. EISENBRAND AND R. WEISMANTEL, *Proximity results and faster algorithms for integer programming using the Steinitz lemma*, in 29th Symposium on Discrete Algorithms (SODA), ACM-SIAM, 2018, pp. 808–816. doi: [10.1137/1.9781611975031.52](https://doi.org/10.1137/1.9781611975031.52).
- [FG01] P. FRAIGNIAUD AND C. GAVOILLE, *Routing in trees*, in 28th International Colloquium on Automata, Languages and Programming (ICALP), F. Orejas, P. G. Spirakis, and J. v. Leeuwen, eds., vol. 2076 of Lecture Notes in Computer Science, Springer, July 2001, pp. 757–772. doi: [10.1007/3-540-48224-5_62](https://doi.org/10.1007/3-540-48224-5_62).
- [FGNW17] O. FREEDMAN, P. GAWRYCHOWSKI, P. K. NICHOLSON, AND O. WEIMANN, *Optimal distance labeling schemes for trees*, in 36th Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM Press, July 2017, pp. 185–194. doi: [10.1145/3087801.3087804](https://doi.org/10.1145/3087801.3087804).
- [FK10a] P. FRAIGNIAUD AND A. KORMAN, *Compact ancestry labeling schemes for XML trees*, in 21st Symposium on Discrete Algorithms (SODA), ACM-SIAM, January 2010, pp. 458–466. doi: [10.1137/1.9781611973075.38](https://doi.org/10.1137/1.9781611973075.38).
- [FK10b] P. FRAIGNIAUD AND A. KORMAN, *An optimal ancestry scheme and small universal posets*, in 42nd Annual ACM Symposium on Theory of Computing (STOC), ACM Press, June 2010, pp. 611–620. doi: [10.1145/1806689.1806773](https://doi.org/10.1145/1806689.1806773).
- [GIP18] D. GONÇALVES, L. ISENMANN, AND C. PENNARUN, *Planar graphs as L-intersection or L-contact graphs*, in 29th Symposium on Discrete Algorithms (SODA), ACM-SIAM, 2018, pp. 172–184. doi: [10.1137/1.9781611975031.12](https://doi.org/10.1137/1.9781611975031.12).
- [GKU16] P. GAWRYCHOWSKI, A. KOSOWSKI, AND P. UZNAŃSKI, *Sublinear-space distance labeling using hubs*, in 30th International Symposium on Distributed Computing (DISC), vol. 9888 of Lecture Notes in Computer Science (ARCoSS), Springer, September 2016, pp. 230–242. doi: [10.1007/978-3-662-53426-7_17](https://doi.org/10.1007/978-3-662-53426-7_17).
- [GL07] C. GAVOILLE AND A. LABOUREL, *Shorter implicit representation for planar graphs and bounded treewidth graphs*, in 15th Annual European Symposium on Algorithms (ESA), L. Arge and E. Welzl, eds., vol. 4698 of Lecture Notes in Computer Science, Springer, October 2007, pp. 582–593. doi: [10.1007/978-3-540-75520-3_52](https://doi.org/10.1007/978-3-540-75520-3_52).
- [GP03] C. GAVOILLE AND D. PELEG, *Compact and localized distributed data structures*, Distributed Computing, 16 (2003), pp. 111–120. doi: [10.1007/s00446-002-0073-5](https://doi.org/10.1007/s00446-002-0073-5), PODC 20-Year Special Issue.
- [GP08] C. GAVOILLE AND C. PAUL, *Optimal distance labeling for interval graphs and related graphs families*, SIAM Journal on Discrete Mathematics, 22 (2008), pp. 1239–1258. doi: [10.1137/050635006](https://doi.org/10.1137/050635006).
- [GU16] P. GAWRYCHOWSKI AND P. UZNAŃSKI, *A note on distance labeling in planar graphs*, Tech. Rep. [1611.06529v1 \[cs.DS\]](https://arxiv.org/abs/1611.06529v1), arXiv, November 2016.
- [KKKP04] M. KATZ, N. A. KATZ, A. KORMAN, AND D. PELEG, *Labeling schemes for flow and connectivity*, SIAM Journal on Computing, 34 (2004), pp. 23–40. doi: [10.1137/S0097539703433912](https://doi.org/10.1137/S0097539703433912).

- [KNR88] S. KANNAN, M. NAOR, AND S. RUDICH, *Implicit representation of graphs*, in 20th Annual ACM Symposium on Theory of Computing (STOC), ACM Press, May 1988, pp. 334–343. doi: [10.1145/62212.62244](https://doi.org/10.1145/62212.62244).
- [Koe36] P. KOEBE, *Kontaktprobleme der konformen abbildung*, Berichte über die Verhandlungen der Sächsische Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse, 88 (1936), pp. 141–164.
- [Pel00] D. PELEG, *Informative labeling schemes for graphs*, in 25th International Symposium on Mathematical Foundations of Computer Science (MFCS), vol. 1893 of Lecture Notes in Computer Science, Springer, August 2000, pp. 579–588. doi: [10.1007/3-540-44612-5_53](https://doi.org/10.1007/3-540-44612-5_53).
- [Pel05] D. PELEG, *Informative labeling schemes for graphs*, Theoretical Computer Science, 340 (2005), pp. 577–593. doi: [10.1016/j.tcs.2005.03.015](https://doi.org/10.1016/j.tcs.2005.03.015).
- [PS19] M. PILIPCZUK AND S. SIEBERTZ, *Polynomial bounds for centered colorings on proper minor-closed graph classes*, in 30th Symposium on Discrete Algorithms (SODA), ACM Press, January 2019, pp. 1501–1520. doi: [10.1137/1.9781611975482.91](https://doi.org/10.1137/1.9781611975482.91).
- [RLDL94] C. J. RHEE, Y. D. LIANG, S. K. DHALL, AND S. LAKSHMIVARAHAN, *Efficient algorithms for finding depth-first and breadth-first search trees in permutation graphs*, Information Processing Letters, 49 (1994), pp. 45–50. doi: [10.1016/0020-0190\(94\)90053-1](https://doi.org/10.1016/0020-0190(94)90053-1).
- [Rot16] N. ROTBART, *New Ideas on Labeling Schemes*, phd thesis, University of Copenhagen, October 2016. http://www.academia.edu/33855491/New_Ideas_on_Labeling_Schemes.
- [RS04] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. XX. Wagner’s conjecture*, Journal of Combinatorial Theory, Series B, 92 (2004), pp. 325–357. doi: [10.1016/j.jctb.2004.08.001](https://doi.org/10.1016/j.jctb.2004.08.001).
- [RT15] L. RODITTY AND R. TOV, *New routing techniques and their applications*, in 34th Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM Press, July 2015, pp. 23–32. doi: [10.1145/2767386.2767409](https://doi.org/10.1145/2767386.2767409).
- [SB77] S. STAHL AND L. W. BEINEKE, *Blocks and the nonorientable genus of graphs*, Journal of Graph Theory, 1 (1977), pp. 75–78. doi: [10.1002/jgt.3190010114](https://doi.org/10.1002/jgt.3190010114).
- [SB97] S. SEVAST’JANOV AND W. BANASZCZYK, *To the Steinitz lemma in coordinate form*, Discrete Mathematics, 169 (1997), pp. 145–152. doi: [10.1016/0012-365X\(94\)00240-J](https://doi.org/10.1016/0012-365X(94)00240-J).
- [Sch89] W. SCHNYDER, *Planar graphs and poset dimension*, Order, 5 (1989), pp. 323–343. doi: [10.1007/BF00353652](https://doi.org/10.1007/BF00353652).
- [Sev78] S. SEVAST’JANOV, *Approximate solution of some problems of scheduling theory*, Metody Diskret. Analiz, 32 (1978), pp. 66–75.
- [Spi03] J. P. SPINRAD, *Efficient Graph Representations*, vol. 19 of Fields Institute Monographs, American Mathematical Society, 2003.
- [Tho86] C. THOMASSEN, *Interval representations of planar graphs*, Journal of Combinatorial Theory, Series B, 40 (1986), pp. 9–20. doi: [10.1016/0095-8956\(86\)90061-4](https://doi.org/10.1016/0095-8956(86)90061-4).

- [TZ01] M. THORUP AND U. ZWICK, *Compact routing schemes*, in 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), ACM Press, July 2001, pp. 1–10. doi: [10.1145/378580.378581](https://doi.org/10.1145/378580.378581).

A Bounded Treewidth Graphs: Technical Details

In this section we explain the proof of Theorem 4. In Section A.1 we explain how property (P1) of the scheme follows immediately from the general strategy borrowed from [GL07], while in Section A.2 we replace the crucial ingredient of the argument from [GL07] in order to achieve property (P2).

A.1 Encoding and decoding

As mentioned, the scheme of Gavaille and Labourel provided in [GL07], without any modifications, already possesses property (P1). Hence, we now recall the reasoning for completeness, and because we will need its understanding for achieving property (P2). Our presentation is a bit simplified compared to that of [GL07], because we choose not to optimize the label length as much as there (e.g. Gavaille and Labourel actually provide an upper bound of $\log n + O(k \log \log (n/k))$ instead of $\log n + O(k \log \log n)$ by a more precise analysis).

First, since the input graph G has treewidth at most k , one can obtain a chordal supergraph G^+ of G on the same vertex set such that G^+ also has treewidth at most k . This can be done as follows: take a tree decomposition of G of width at most k and turn every bag into a clique. Since for fixed k such a tree decomposition can be computed in linear time [Bod96], G^+ can be computed in linear time.

Next, it is well-known that since G^+ is chordal and of treewidth at most k , in linear time we can compute an orientation \vec{G} of G^+ such that every vertex u has at most k out-neighbors in \vec{G} , and moreover u together with those out-neighbors form a clique in G^+ . For every $u \in V(G)$, let K_u be the set consisting of u and its out-neighbors in \vec{G} .

The key idea of the approach of Gavaille and Labourel is to compute a *bidecomposition* of the graph G^+ , which is a notion roughly resembling tree decompositions, but actually quite different.

Definition 2. A bidecomposition of a graph H is a pair (T, α) , where T is a binary rooted tree and α maps vertices H to nodes of T , so that for every edge uv of H , $\alpha(u)$ and $\alpha(v)$ are related.

As proved in [GL07], graphs of bounded treewidth admit bidecompositions with small parts. This is the key combinatorial ingredient of the proof.

Lemma 7 (cf. Lemma 1 in [GL07]). Let G be an n -vertex graph of treewidth at most k . Then there exists a bidecomposition (T, α) of G satisfying the following:

- (A1) $|\alpha^{-1}(x)| = O(k \log n)$ for every node x of T ; and
- (A2) T has depth at most $\log n$.

Moreover, for every fixed k , given G such a bidecomposition can be constructed in time $O(n \log n)$.

We apply Lemma 7 to the graph G^+ , thus getting a suitable bidecomposition (T, α) . Based on this, a labeling is constructed as follows.

Consider any $u \in V(G)$. Since K_u is a clique in G^+ , it follows that nodes $\{\alpha(v)\}_{v \in K_u}$ are pairwise related. Hence, there exists a path P_u in T starting at the root that contains all nodes $\alpha(v)$ for $v \in K_u$. The second endpoint of P_u is the deepest among nodes $\{\alpha(v)\}_{v \in K_u}$. Let P'_u be the prefix of P_u from the root of T to $\alpha(u)$.

For each node x of T fix an arbitrary enumeration of $\alpha^{-1}(x)$ using index taken from $[0, |\alpha^{-1}(x)|)$. Now, the identifier of vertex u consists of the following pieces of information:

1. The encoding of the path P'_u as a bit string of length $|V(P'_u)| - 1$ that encodes, for consecutive non-root vertices of P'_u , whether they are left or right children.
2. The index of u within $\alpha^{-1}(\alpha(u))$.
3. The depth of $\alpha(u)$ in T .

Since T has depth at most $\log n$ and $|\alpha^{-1}(x)| = O(k \log n)$ for every node x of T , we conclude that the identifier has total length $\log n + \log k + O(\log \log n)$. In addition to the identifier, the label of u contains the following pieces of information:

1. Encoding of the suffix of P_u that is not contained in P'_u ; this, together with the information from the identifier, adds up to the encoding of P_u .
2. For every $v \in K_u \setminus \{u\}$, the depth of $\alpha(v)$ in T , the index of v within $\alpha^{-1}(\alpha(v))$, and whether the edge uv belongs to $E(G)$ (it may belong to $E(G^+) \setminus E(G)$).

As shown in [GL07], the above information, together with the identifier, can be encoded in $\log n + O(k \log \log n)$ bits, resulting in the promised upper bound on the label length.

It is now straightforward to see that from the label of u one can derive the identifiers of the out-neighbors of u in G^+ . Indeed, for every $v \in K_u \setminus \{u\}$ the depth of $\alpha(v)$ and the index of v in $\alpha^{-1}(\alpha(v))$ are directly stored in the label of u , while the encoding of the path P'_v can be obtained by taking the encoding of P_u and trimming it to the prefix of length equal to the depth of $\alpha(v)$. With every such out-neighbor v we have also stored the information whether the edge uv is contained in G , or it was added when modifying G to G^+ . Hence, given the label $\lambda(u)$ we can compute a set of at most k identifiers of neighbors of u , which is a suitable set $\Gamma(\lambda(u))$. This proves property (P1).

A.2 Saving on labels of a small set of vertices

We now explain how the general approach of Gavaille and Labourel [GL07], presented in the previous section, can be amended to achieve property (P2) as well. The difference is that we replace the usage of Lemma 7 with the following Lemma 8.

Lemma 8. *Let G be an n -vertex graph of treewidth at most k and $S \subseteq V(G)$. Then there exists a bidecomposition (T, α) of G satisfying the following:*

- (B1) $|\alpha^{-1}(x)| = O(k \log n)$ for every node x of T ;
- (B2) T has depth at most $\log n + O(1)$; and
- (B3) for every $u \in S$, $\alpha(u)$ is at depth at most $\log |S| + O(1)$ in T .

Moreover, for every fixed k , given G and S such a decomposition can be constructed in polynomial time, with the degree of the polynomial independent of k .

Consider the set Q of prescribed vertices as in property (P2), and apply Lemma 8 to G^+ with

$$S = \bigcup_{u \in Q} K_u.$$

We have $|S| \leq (k+1) \cdot |Q|$. Hence, in the notation of the previous section, for every $u \in Q$ we have that P_u has at most $\log |S| + O(1) = \log |Q| + O(\log k)$ nodes, while for every other vertex u we have that P_u

has at most $\log n + O(1)$ nodes. Plugging this into the analysis of the previous section gives the desired bounds on the lengths of labels in the constructed labeling. Note that thus, property (P1) still holds, while property (P2) is achieved.

We are left with proving Lemma 8. We would again like to stress that this is *not* a simple modification of the proof of Lemma 7 presented in [GL07]. The general idea is to recursively decompose the graph, where at each step we use a separator of size $O(k \log n)$ to split the graph into two parts, each containing (roughly) at most half of the remaining vertices and at most half of the remaining vertices of S . In [GL07] only the first objective – halving the total number of vertices – was necessary, and this was relatively easy to achieve using a separator of size $O(k \log n)$. However, the strategy used in [GL07] does not generalize to achieving both objectives at the same time. Hence, our splitting step is based on a completely different argument.

PROOF (OF LEMMA 8). We focus on proving the existential statement and discuss the algorithmic aspects at the end.

For a graph H and a weight function $\omega: V(H) \rightarrow \mathbb{R}_{\geq 0}$, we write $\omega(H) = \sum_{u \in V(H)} \omega(u)$. We will use the following well-known claim.

Claim 4 (see Lemma 7.19 of [CFK⁺15]). *Let H be a graph of treewidth at most k and let $\omega: V(H) \rightarrow \mathbb{R}_{\geq 0}$ be a weight function on the vertices of H . Then there exists a set $Z \subseteq V(H)$ of size at most $k + 1$ such that for every connected component C of $H - Z$ we have $\omega(C) \leq \frac{1}{2}\omega(H)$.*

First, we will need the following generalization of Claim 4.

Claim 5. *Let $\varepsilon > 0$, let H be a graph of treewidth at most k , and let $\omega: V(H) \rightarrow \mathbb{R}_{\geq 0}$ be a weight function on the vertices of H . Then there exists a set $Z \subseteq V(H)$ of size $O(\varepsilon^{-1}k)$ such that for every connected component C of $H - Z$ we have $\omega(C) \leq \varepsilon \cdot \omega(H)$.*

PROOF. Consider the following iterative procedure. Start with $Z = \emptyset$. As long as $H - Z$ contains a component C satisfying $\omega(C) > \varepsilon \cdot \omega(H)$, pick any such component C and apply Claim 4 to it. This yields a suitable set Z_C of size at most $k + 1$ such that every connected component C' of $C - Z_C$ satisfies $\omega(C') \leq \frac{1}{2}\omega(C)$. Then add Z_C to Z and continue.

The procedure eventually stops, yielding a set Z satisfying the requested condition and of size at most $(k + 1) \cdot \ell$, where ℓ is the number of iterations. Therefore, it suffices to show that $\ell = O(\varepsilon^{-1})$.

We partition the iterations according to how the component C considered in the iteration is split:

- **Leaf iterations:** for every component C' of $C - Z_C$ we have $\omega(C') \leq \varepsilon \cdot \omega(H)$.
- **Unary iterations:** there is exactly one component C' of $C - Z_C$ satisfying $\omega(C') > \varepsilon \cdot \omega(H)$.
- **Binary iterations:** there are at least two components C' of $C - Z_C$ satisfying $\omega(C') > \varepsilon \cdot \omega(H)$.

For leaf iterations, observe that the connected component of $C - Z_C$ are not touched by the procedure any more, and survive as connected components of $H - Z$ till the end. Hence, all the components C considered in all leaf iterations are pairwise disjoint. Since each of them satisfies $\omega(C) > \varepsilon \cdot \omega(H)$, we conclude that the total number of leaf iterations is at most ε^{-1} .

For unary iterations, a similar argument applies. If C' is the unique component of $C - Z_C$ satisfying $\omega(C') > \varepsilon \cdot \omega(H)$, then we also have $\omega(C') \leq \frac{1}{2}\omega(C)$, which implies that $\sum_{C''} \omega(C'') \geq \frac{1}{2}\omega(C) \geq \varepsilon/2 \cdot \omega(H)$, where the sum goes over all components C'' of $C - Z_C$ that are different from C' . Since these

components C'' are not touched any more by the procedure, they are pairwise disjoint for different unary iterations, yielding an upper bound of $2\varepsilon^{-1}$ on the total number of unary iterations.

Finally, to bound the number of binary iteration, we may imagine the run of the procedure as a rooted forest labeled with components to which the iterations are applied: the children of a node labeled by a component C are labeled with components C' of $C - Z_C$ satisfying $\omega(C') > \varepsilon \cdot \omega(H)$, to which the procedure was further applied. In this forest, leaf iterations correspond to leaves, unary iterations correspond to nodes with one child, and binary iterations correspond to nodes with at least two children. Since the number of nodes with at least two children in such a forest is always smaller than the number of leaves, we conclude that the number of binary iterations is smaller than the number of leaf iterations, which in turn is upper bounded by ε^{-1} .

All in all, we conclude that the total number of iterations is bounded by $4\varepsilon^{-1}$, resulting in a bound of $4\varepsilon^{-1} \cdot (k + 1) = O(\varepsilon^{-1}k)$ on the size of Z . \square

We now use Claim 5 to find separators that are suited for constructing a bidecomposition. The idea is that after removing the separator, we need to be able to group the remaining components into two parts that are roughly balanced: both in terms of the number of vertices and in terms of the number of vertices of Q . We prefer to put this condition in an abstract way, using two weight functions.

Claim 6. *Let $\varepsilon > 0$, let H be a graph of treewidth at most k , and let $\omega_1, \omega_2: V(H) \rightarrow \mathbb{R}_{\geq 0}$ be two weight functions on the vertices of H . Then there exists a partition (A, X, B) of $V(G)$ with the following properties:*

- *there is no edge with one endpoint in A and second in B ;*
- *$|X| = O(\varepsilon^{-1} \cdot k)$; and*
- *$\omega_t(A) \leq (1/2 + \varepsilon) \cdot \omega_t(H)$ and $\omega_t(B) \leq (1/2 + \varepsilon) \cdot \omega_t(H)$, for $t \in \{1, 2\}$.*

PROOF. Without loss of generality we assume that $\frac{1}{2\varepsilon}$ is an integer, in particular $\varepsilon < \frac{1}{2}$.

Apply Claim 5 to H with weight functions $\omega_1(\cdot)$ and $\omega_2(\cdot)$, yielding suitable separators $Z_1, Z_2 \subseteq V(H)$. Let $X = Z_1 \cup Z_2$. Then $|X| = O(\varepsilon^{-1} \cdot k)$ and

$$\omega_t(C) \leq \varepsilon \cdot \omega_t(H) \quad \text{for each } t \in \{1, 2\} \text{ and every connected component } C \text{ of } H - X .$$

It therefore suffices to prove that the connected components of $H - X$ can be partitioned into two groups A and B so that the third condition is satisfied.

By scaling the weight functions we may assume that $\omega_1(H - X) = \omega_2(H - X) = 1$ (if $\omega_t(H - X) = 0$ for any $t \in \{1, 2\}$, then the third condition is trivially satisfied for t). Consider the following multiset W of vectors in \mathbb{R}^2 : for each connected component C of $H - X$ add to W the vector

$$a_C = (\omega_1(C), \omega_2(C)),$$

and then add to W exactly $1/\varepsilon$ copies of the following vector:

$$b = (-\varepsilon, -\varepsilon) .$$

Observe that the sum of vectors in W is equal to $(0, 0)$ and that the ∞ -norm of each of them is at most ε . Therefore, by Steinitz Lemma [Sev78, SB97] we conclude that there exists a permutation v_1, v_2, \dots, v_p of vectors in W , where p is the cardinality of W , such that

$$\|v_1 + \dots + v_j\|_{\infty} \leq 2\varepsilon \quad \text{for each } j \in \{0, \dots, p\} . \tag{1}$$

Let $q \leq p$ be such that exactly $\frac{1}{2\varepsilon}$ among vectors v_1, \dots, v_q are equal to b ; recall that we assumed that $\frac{1}{2\varepsilon}$ is an integer, so such q exists. Let $A \subseteq V(G)$ consist of the vertex sets of those connected components of $H - X$ for which a_C is among v_1, \dots, v_q , and let $B \subseteq V(G)$ consist of the vertex sets of the remaining components of $H - X$. Then by (1) and the definition of q we conclude that

$$|\omega_t(A) - 1/2| \leq 2\varepsilon \quad \text{for each } t \in \{1, 2\} .$$

This implies that

$$\omega_t(A) \leq 1/2 + 2\varepsilon \leq (1/2 + 2\varepsilon) \cdot |V(H)| \quad \text{for each } t \in \{1, 2\} ,$$

and similarly for B . We conclude by rescaling ε by a multiplicative factor of 2 throughout the proof. \square

We are finally ready to build the requested bidecomposition. Let $n = |V(G)|$ and fix $\varepsilon = 1/\log n$. W.l.o.g. assume that $\varepsilon \leq 1/8$, for otherwise G is of constant size. Define weight functions on vertices of G as follows: $\omega_1(u) = 1$ for each vertex u , and $\omega_2(u) = 1$ for each $u \in S$ and $\omega_2(u) = 0$ for each $u \notin S$. The bidecomposition is constructed using the following recursive procedure which for $R \subseteq V(G)$ constructs a bidecomposition of $G[R]$; we apply it initially to $R = V(G)$.

1. If R is empty, terminate and return an empty bidecomposition.
2. Otherwise, apply Claim 6 to $H = G[R]$ with weight functions $\omega_1(\cdot)$ and $\omega_2(\cdot)$ restricted to R . This yields a partition (A, X, B) of R . Apply the procedure recursively to A and to B in place of R , yielding bidecompositions of $G[A]$ and $G[B]$, respectively. Return a bidecomposition of $G[R]$ obtained by creating a root r , mapping all vertices of X to r , and attaching the bidecompositions of $G[A]$ and $G[B]$ as children of r .

Let (T, α) be the bidecomposition of G obtained in this manner. As $\varepsilon = 1/\log n$, property (B1) is clear from the construction. Therefore, we focus on proving properties (B2) and (B3).

Consider any root-to-leaf path in T and let

$$V(G) = R_0 \supseteq R_1 \supseteq R_2 \supseteq \dots \supseteq R_d$$

be the consecutive sets R considered by the procedure constructing (T, α) along this path. By the construction we have

$$|R_i| \leq (1/2 + \varepsilon)^i \cdot n \quad \text{for each } i \in \{0, \dots, d\} .$$

Since the procedure stops when R becomes empty, we have that $|R_d| \geq 1$, implying that

$$(1/2 + \varepsilon)^d \cdot n \geq 1$$

or equivalently

$$d \leq -\log_{1/2+\varepsilon} n = \frac{\log n}{1 + \log\left(\frac{1}{1+2\varepsilon}\right)} .$$

Now observe that

$$\log\left(\frac{1}{1+2\varepsilon}\right) = \log\left(1 - \frac{2\varepsilon}{1+2\varepsilon}\right) \geq -\frac{4\varepsilon}{1+2\varepsilon} \geq -4\varepsilon .$$

Here, the first inequality follows from $\log(1 - x) \geq -2x$ for $x \in [0, 1/2]$, which in turn follows from the concavity of function $t \mapsto \log t$. Therefore, we conclude that

$$d \leq \frac{\log n}{1 - 4\varepsilon} \leq \log n \cdot (1 + 8\varepsilon) = \log n + 8,$$

where the second inequality follows from $1 \leq (1 - 4\varepsilon)(1 + 8\varepsilon)$ being true for $\varepsilon \leq 1/8$. We conclude that the height of T is at most $\log n + 8$, so property (B2) is verified.

The proof of property (B3) is analogous: instead of any root-to-leaf path in T , we consider any path from the root to any node x satisfying $\alpha^{-1}(x) \cap S \neq \emptyset$. This concludes the proof of the existential statement.

We now discuss the algorithmic aspects. The standard proof of Claim 4, see e.g. [CFK⁺15], shows that some bag of a tree decomposition of width k of H is a suitable set Z . Since for fixed k such a decomposition can be found in linear time [Bod96], the proof of Claim 4 yields a linear-time algorithm for finding Z . The proof of Claim 5 amounts to applying Claim 4 $O(\varepsilon^{-1}) = O(\log n)$ times, so this yields an $O(n \log n)$ -time algorithm for this claim. We need to be careful with Claim 6: after applying Claim 5 twice, in $O(n \log n)$ time, we consider a permutation of vectors from W that is given by the Steinitz Lemma, which is a priori a non-constructive argument. However, there are known proofs of the Steinitz Lemma that yield polynomial-time algorithms for finding a suitable permutation via iterative application of linear programming, see e.g. the discussion in [EW18]. Hence, the conclusion of Claim 6 can be also found in polynomial time for fixed k . Finally, the construction of the final bidecomposition amounts to applying (the algorithm of) Claim 6 a polynomial number of times. \square

B Graphs of Bounded Genus

In this section we prove the following strengthening of Theorem 6.

Theorem 9. *For every fixed $g \in \mathbb{N}$, graphs of Euler genus at most g admit a labeling scheme of length $\frac{4}{3} \log n + O((g + 1) \log \log n)$. The Encoder runs in polynomial time, with the degree of the polynomial independent of g , and the Decoder runs in constant time.*

For the rest of this section we fix $g \in \mathbb{N}$ and let \mathcal{D} be the class of graphs of Euler genus at most g . The proof follows from applying the same approach as that of Theorem 6, hence we only discuss how we modify the parts of the argumentation, where the planarity of the input graph was used. A close inspection of the proof of Theorem 6 shows that there are three such places:

1. In the very beginning we used planarity when assuming that the input graph is connected.
2. In Claim 2 we used planarity to argue that the graph G_1 has treewidth 6.
3. In Claim 3 we used planarity to obtain a labeling for G_2 by applying the labeling scheme for planar graphs of small diameter, i.e., Theorem 5.

We now fix these parts in order.

Making the graph connected. In the beginning of the proof of Theorem 6 we assumed that the input graph G is connected. The argument was as follows: given a disconnected G , one can make it connected while not spoiling planarity by adding one vertex with a single neighbor in each connected component of

G , and applying the scheme to the graph G' obtained in this way. While this operation preserves planarity (i.e. G' is planar as well), a priori it is not clear whether it also preserves belonging to \mathcal{D} . However, this is actually the case.

Lemma 10. *Let G be a graph and construct G' from G by adding a new vertex and making it adjacent to one arbitrary vertex from each connected component of G . Then the Euler genera of G and G' are equal.*

PROOF. This is a direct corollary of the following result of Stahl and Beineke [SB77]: the Euler genus of a graph is equal to the sum of Euler genera of its 2-connected components. \square

This means that in the proof of Theorem 9 we may assume that the input graph is connected in the same way as in the proof of Theorem 6.

Treating G_1 . In the proof of Claim 2 we used Lemma 1 to argue that G_1 has treewidth at most 6, because every connected component of G_1 is a subgraph of a connected planar graph of radius at most 2. This planar graph was obtained from G by means of minor operations, hence it was clear that it was planar.

In the setting of Theorem 9, we know that $G \in \mathcal{D}$ and since \mathcal{D} is closed under taking minors, in the same way we also conclude that every connected component of G_1 is a subgraph of a connected graph from \mathcal{D} of radius at most 2. Then we can substitute the usage of Lemma 1 with the following well-known generalization.

Lemma 11 ([Epp00]). *A connected graph of genus at most g and radius ρ has treewidth $O(g\rho)$.*

It then follows that G_1 has treewidth $O(g)$. Hence, the labeling scheme for G_1 , obtained using Theorem 3, will use labels of length $\log(n/d) + O((g+1)\log\log n)$.

Treating G_2 . Finally, in the proof of Theorem 6 we argued that we can obtain a labeling of G_2 using labels of length $\log n + \log d + O(\log\log n) = \frac{4}{3}\log n + O(\log\log n)$, where the vertices that G_2 has in common with G_1 receive shorter labels: of length $\frac{2}{3}\log n + O(\log\log n)$. For this, we argued that G_2 is a subgraph of a connected planar graph H on at most $2n$ vertices where, according to Claim 3, every vertex is at distance at most d from a fixed vertex s , and moreover vertices in $V(G_1)$ appear only at distance 1 or d . Then Theorem 5 was applied to H in order to get a labelling for G_2 .

The planarity of H was argued as follows: H can be obtained from a sequence of graphs $R''_0, R''_1, R''_2, \dots$ by taking their disjoint union, picking one vertex from each and identifying the picked vertices into one vertex s . Here, in the notation of the proof of Theorem 5 we have $R''_i = R'_i$ for $i \geq 1$, and R''_0 is obtained from R'_0 by attaching a (possibly zero-length) path to the vertex r . As each R''_i is planar, because it can be obtained from G either a sequence of contractions or (in case of R''_0) attaching a path to a vertex, it follows that H is planar as well.

We now examine how this approach can be lifted to the setting of Theorem 6. As \mathcal{D} is minor-closed and closed under attaching paths to vertices, we also conclude that graphs $R''_0, R''_1, R''_2, \dots$, defined in the same way as above, belong to \mathcal{D} . However, the issue is that the graph H obtained from them as described in the previous paragraph does not necessarily belong to \mathcal{D} .

Instead, let us define a class of graph $\widehat{\mathcal{D}}$ that comprises all graphs J which can be obtained as follows. Consider any sequence of connected graphs $J_1, \dots, J_p \in \mathcal{D}$ and vertices u_1, \dots, u_p , where $u_i \in V(J_i)$

for $i = 1, \dots, p$. Then construct J by taking the disjoint union of J_1, \dots, J_p and identifying all vertices u_1, \dots, u_p into one vertex u , called further the *corner* of J . Then the graph H from the previous paragraph belongs to $\widehat{\mathcal{D}}$, with s being the corner. We conclude that, in order to be able to apply exactly the same reasoning, we need to prove the following generalization of Theorem 5 that will be used in its place.

Theorem 12. *Graphs from $\widehat{\mathcal{D}}$ admit a labeling scheme of length at most $\log n + \log d + O(g + \log \log n)$, where n is the vertex count of the graph and d is the maximum distance from any vertex to the corner of the input graph. The Encoder runs in polynomial time and the Decoder runs in constant time.*

Moreover, if the graph is provided together with a vertex subset Q , then the Encoder may assign to the vertices of Q labels of length at most $\log |Q| + \log d + O(g + \log \log n)$.

Let us note that, similarly as in the proof of Theorem 5, we use the trick expressed in Remark 1 to reduce the $\log d$ summand to $O(1)$ for vertices of G_2 that also belong to G_1 .

Proving Theorem 12. Again, the proof of Theorem 12 follows the same approach as that of Theorem 5. We will rely on generalizations of Theorem 2 to graphs of bounded Euler genus proved by Dujmović et al. in [DJM⁺19]. For this, we need several definitions from this work and results from this work.

Definition 3. *For a graph G , a layering of G is a partition of its vertex set into layers (V_0, V_1, V_2, \dots) such that every edge of G connects two vertices from the same layer or from two consecutive layers. A BFS layering of a connected graph G is obtained by taking $V_i = \{v : \text{dist}(u, v) = i\}$ for some fixed vertex u of G .*

Definition 4. *For a partition \mathcal{P} of $V(G)$, the layered width of \mathcal{P} with respect to a layering (V_0, V_1, V_2, \dots) is the least integer ℓ such that $|A \cap V_i| \leq \ell$ for each $A \in \mathcal{P}$ and $i \in \mathbb{N}$.*

Then the generalization of Theorem 2 we are going to use can be phrased as follows.

Theorem 13 (Theorem 20 of [DJM⁺19]). *Every graph $G \in \mathcal{D}$ has a partition \mathcal{P} with layered width at most $\max\{2g, 1\}$ such that G/\mathcal{P} has treewidth at most 9. Moreover, there is such a partition for every given BFS layering of G , and such a partition can be computed in polynomial time with the degree independent of g .*

Note that the algorithmic statement is not provided explicitly in [DJM⁺19], but is discussed in Section 10 there. We then have the following easy lift of Theorem 13 to $\widehat{\mathcal{D}}$.

Lemma 14. *Consider any graph $G \in \widehat{\mathcal{D}}$ and let (V_0, V_1, V_2, \dots) be its BFS layering obtained by running the BFS starting from the corner of G . Then G has a partition \mathcal{P} that has layered width at most $\max(2g, 1)$ w.r.t. (V_0, V_1, V_2, \dots) such that G/\mathcal{P} has treewidth at most 10. Moreover, such a partition can be computed in polynomial time with the degree independent of g .*

PROOF. Let $G_1, \dots, G_p \in \mathcal{D}$ and u_1, \dots, u_p be such that G is obtained by taking the disjoint union of connected graphs G_1, \dots, G_p and identifying vertices u_1, \dots, u_p , where $u_j \in V(G_j)$, into one vertex u , which is the corner of G . For $j \in \{1, \dots, p\}$, let (V_0^j, V_1^j, \dots) be the BFS layering of G_j obtained by running a BFS from u_j . Then the BFS layering (V_0, V_1, V_2, \dots) satisfies

$$V_0 = \{u\} \quad \text{and} \quad V_i = \bigcup_{j=1}^p V_i^j \quad \text{for } i \geq 1.$$

Apply Theorem 13 to each G_j with BFS layering (V_0^j, V_1^j, \dots) , thus obtaining a partition \mathcal{P}_j of $V(G_j)$ that has layered width at most $\max\{2g, 1\}$ w.r.t. (V_0^j, V_1^j, \dots) , and such that G_j/\mathcal{P}_j has treewidth at most 9. Let \mathcal{P}'_j be the vertex partition of $G_j - u_j$ obtained from \mathcal{P}_j by removing u_j from the unique part that it belongs to. Finally, let

$$\mathcal{P} = \{u\} \cup \bigcup_{j=1}^p \mathcal{P}'_j.$$

Then it is straightforward to see that \mathcal{P} is a partition of the vertex set of G that has layered width at most $\max\{2g, 1\}$ w.r.t. (V_0, V_1, \dots) . Moreover, G/\mathcal{P} is a subgraph of the graph obtained by adding a universal vertex to disjoint union of G_j/\mathcal{P}_j for $j \in \{1, \dots, p\}$, hence G/\mathcal{P} has treewidth at most 10. \square

With all the tools in place, we can prove Theorem 12.

PROOF (OF THEOREM 12). The proof is analogous to that of Theorem 5, where instead of using Theorem 2 we use Lemma 14 and we modify the scheme as follows. In the labeling scheme of Theorem 5, we used the fact that every part in the considered partition \mathcal{P} has at most one vertex in common with every layer of the fixed BFS tree F , because the parts of \mathcal{P} were columns in F . This allowed us to use an adjacency code consisting of $8 \cdot 3 \cdot 1$ bits, where the consecutive numbers correspond to the treewidth G/\mathcal{P} (which upper bounds the number of identifiers in the set returned by $\Gamma(\cdot)$), the number of adjacent layers (previous, same, and next), and the number of vertices that any part of \mathcal{P} may have in common with any layer. Thus, in the current setting we can use an adjacency code consisting of $10 \cdot 3 \cdot \max\{2g, 1\} = O(g)$ bits in the same way. \square

We can conclude the proof of Theorem 9 by summarizing the bounds on the lengths of labels:

- Vertices of $V(G_2) \setminus V(G_1)$ receive labels of length $\log n + \log d + O(g + \log \log n)$;
- Vertices of G_1 receive labels composed of two sub-labels: one of length $\log(n/d) + O(g + \log \log n)$ inherited from G_1 , and one of length $\log(n/d) + O((g+1) \log \log n)$ inherited from G_2 .

Therefore, for $d = \Theta(n^{1/3})$ the label lengths are bounded by $\frac{4}{3} \log n + O((g+1) \log \log n)$, as claimed.