Treewidth is NP-Complete on Cubic Graphs

Hans L. Bodlaender[a] Édouard Bonnet[b] Lars Jaffke[c]
Dušan Knop[d] Paloma T. Lima[e] Martin Milanič[f]
Sebastian Ordyniak[g] Sukanya Pandey[h] Ondřej Suchý[i]

Submitted: Jul 17, 2024; Accepted: May 7, 2025; Published: Aug 22, 2025

Abstract

In this paper, we show that Treewidth is NP-complete for cubic graphs, thereby improving the result by Bodlaender and Thilikos from 1997 that Treewidth is NP-complete on graphs with maximum degree at most 9. We add a new and simpler proof of the NP-completeness of treewidth, and show that Treewidth remains NP-complete on subcubic induced subgraphs of the infinite 3-dimensional grid, and on cubic line graphs.

Mathematics Subject Classifications: 68Q25,68R10,05C85,05C83,05C78

1 Introduction

Treewidth is one of the most studied graph parameters, with many applications for both theoretical investigations as well as for applications. The problem of determining the treewidth of a given graph, and finding a corresponding tree decomposition, single-handedly led to a plethora of studies, including exact algorithms, algorithms for special graph classes, approximations, upper and lower bound heuristics, parameterised algorithms, and more. In this paper, we look at the basic problem Treewidth: decide, for a given graph G and integer k, whether the treewidth of G is at most k.

In 1987, Arnborg, Corneil and Proskurowski [1] showed this problem to be NP-complete; their proof also gives NP-completeness on co-bipartite graphs. As the treewidth of a graph (without parallel edges) does not change under subdivision of edges, it

[a] Utrecht University, The Netherlands (h.l.bodlaender@uu.nl).
[b] LIP, ENS Lyon, France (edouard.bonnet@ens-lyon.fr)
[c] NHH Norwegian School of Economics, Norway (lars.jaffke@uib.no).
[d] Czech Technical University in Prague, Czech Republic (dusan.knop@fit.cvut.cz).
[e] IT University of Copenhagen, Denmark (palt@itu.dk).
[f] FAMNIT and IAM, University of Primorska, Koper, Slovenia (martin.milanic@upr.si).
[g] University of Leeds, U.K. (sordyniak@gmail.com).
[h] RWTH Aachen, Germany (pandey@algo.rwth-aachen.de).
[i] Czech Technical University in Prague, Czech Republic (ondrej.suchy@fit.cvut.cz).

easily follows and is well known that TREEWIDTH is NP-complete on bipartite graphs. In 1997, Bodlaender and Thilikos [6] modified the construction of Arnborg, Corneil and Proskurowski [1] and showed that TREEWIDTH remains NP-complete if we restrict the inputs to graphs with maximum degree 9. In this paper, we sharpen this bound of 9 to 3. Our proof uses a simple transformation, whose correctness follows from well-known facts about treewidth and simple insights. We also give a new simple proof of the NP-completeness of TREEWIDTH on arbitrary (and on co-bipartite) graphs. We obtain a number of corollaries of the results, in particular NP-completeness of TREEWIDTH on d-regular graphs for each fixed $d \geqslant 3$, for graphs that can be embedded in a 3-dimensional grid, and for cubic line graphs. (A graph is *cubic* if each vertex has degree 3.) Very recently, Bonnet [7] obtained a new NP-completeness proof for treewidth of general graphs; this proof also gives an inapproximability result (assuming P $\neq$ NP), and a $2^{\Omega(n)}$ lower bound (assuming the Exponential Time Hypothesis) for TREEWIDTH.

Our techniques are based on the techniques in [1] and [6] with streamlined and simplified arguments, and some additional new but elementary ideas. As a starting point for the reductions, we use the NP-complete problems CUTWIDTH on cubic graphs and PATHWIDTH; the NP-completeness proofs for these were given by Monien and Sudborough [15] in 1987.

Pathwidth is a well studied graph parameter, related to treewidth. There exist several independent NP-completeness proofs for PATHWIDTH, using different equivalent characterisations of the pathwidth of graphs [1, 10, 13, 16]. Monien and Sudborough [15] showed that PATHWIDTH is NP-complete for graphs of maximum degree 3, using the terminology of vertex separation; as shown by Kinnersley [11], this is equivalent to pathwidth. In this paper, we observe as corollary that the PATHWIDTH problem stays NP-complete for cubic line graphs.

This paper is organised as follows. In Section 2, we give basic definitions and some well-known results on treewidth. In Section 3, we give a new simple proof of the NP-completeness of TREEWIDTH on co-bipartite graphs that uses an elementary transformation from pathwidth. Section 4 gives our main result: NP-completeness for TREEWIDTH on cubic graphs. In Section 5, we derive as consequences some additional NP-completeness results: on d-regular graphs for each fixed d; on graphs that can be embedded in a 3-dimensional grid, and on cubic line graphs. Some final remarks are made in Section 6.

2 Definitions and preliminaries

Throughout the paper, we denote the number of vertices of the graph G by n. All graphs considered in this paper are undirected. A graph G is d-regular if each vertex has degree d. We say that a graph G is *cubic* if G is 3-regular. If each vertex of G has degree at most 3, we say that G is *subcubic*. All numbers considered are assumed to be integers, and an interval $[a, b]$ denotes the set of integers $\{a, a+1, a+2, \ldots, b-1, b\}$. Furthermore, for a positive integer a, we denote by $[a]$ the interval $[1, a]$. A graph G is a *minor* of a graph H if G can be obtained from H by zero or more vertex deletions, edge deletions, and edge contractions. For a graph G and a set of vertices $A \subseteq V(G)$, we write $G + \mathit{clique}(A)$ for the graph obtained by adding an edge between each pair of distinct non-adjacent vertices in A, i.e. by turning A into a clique.

A *tree decomposition* of a graph G is a pair (T, β) such that T is a tree and β is a mapping assigning each node x of T to a *bag* $\beta(x) \subseteq V(G)$, satisfying the following conditions: every vertex of G belongs to some bag, for every edge of G there exists a bag containing both endpoints of the edge, and for every vertex of G, the set of nodes x of T such that $v \in \beta(x)$ induces a connected subtree of T. The *width* of a tree decomposition (T, β) is the maximum, over all nodes x of T, of the value of $|\beta(x)| - 1$. The *treewidth* of a graph G, denoted $\texttt{tw}(G)$, is the minimum width of a tree decomposition of G. Path decompositions and pathwidth (denoted $\texttt{pw}(G)$) are defined analogously, but with the additional requirement that the tree T is a path.

We use a number of well-known and easy to observe facts about treewidth and tree decompositions.

Lemma 1 (Folklore)**.** *Let G be a graph, and (T, β) a tree decomposition of width k of G. Then the following statements hold.*

1. *Let W be a clique in G. Then, there is a node x of T with $W \subseteq \beta(x)$.*
2. *Assume that $v, w \in V(G)$, $\{v, w\} \notin E(G)$. If there is a node x of T with $v, w \in \beta(x)$, then (T, β) is a tree decomposition of width k of the graph obtained by adding the edge $\{v, w\}$ to G.*
3. *Assume that $W \subseteq V(G)$. Then, there is a node x in T such that when we remove $\beta(x)$ and all incident edges from G, then each connected component of G contains at most $|V(G)|/2$ vertices of W.*
4. *Let y be a leaf of T, with parent y'. If $\beta(y) \subseteq \beta(y')$, then removing y with its bag from the tree decomposition (T, β) yields another tree decomposition of G of width at most k.*
5. *If H is a minor of G, then $\texttt{tw}(H) \leqslant \texttt{tw}(G)$, and $\texttt{pw}(H) \leqslant \texttt{pw}(G)$.*

A graph G is *co-bipartite* if $V(G) = A \cup B$ with A a clique and B a clique (that is, the complement of G is bipartite). The following fact is also well known, and follows implicitly from the proofs of Arnborg et al. [1]. For completeness, we give a proof here.

Lemma 2 (See, e.g. [1])**.** *Let G be a co-bipartite graph, with $V(G) = A \cup B$ where A and B are cliques. Then:*

1. *$\texttt{tw}(G) = \texttt{pw}(G)$.*
2. *G has a path decomposition (P, β) with width equal to $\texttt{tw}(G)$ such that $A \subseteq \beta(p_1)$ and $B \subseteq \beta(p_r)$, where p_1 and p_r are the two endpoints of P.*

Proof. Let (T, β) be a tree decomposition of G of width $\mathtt{tw}(G)$. By Lemma 1(1), there is a node x in T with $A \subseteq \beta(x)$, and a node y in T with $B \subseteq \beta(y)$. Let P be the path from x to y in T.

If there are nodes in T that do not belong to P, then we can apply the following step. At least one such node must be a leaf of T. Take a leaf z of T such that z is not in P. Let z' be the parent (i.e., unique neighbour) of z in T. For each $v \in A \cap \beta(z)$, it holds that $v \in \beta(z')$ as z' is on the path from z to x, and for each $v \in B \cap \beta(z)$, it holds that $v \in \beta(z')$ as z' is on the path from z to y. So, by Lemma 1(4), we can remove z from T and obtain another tree decomposition of G. Repeating this step as long as possible gives the desired result. □

The *vertex separation number* of a graph G is denoted by $\mathtt{vsn}(G)$ and defined as the minimum, over all orderings $\sigma = (v_1, \ldots, v_n)$ of the vertex set of G, of the maximum, over all $i \in \{1, \ldots, n\}$, of the number of vertices v_j such that $j > i$ and v_j has a neighbour in $\{v_1, \ldots, v_i\}$. Kinnersley proved the following characterisation of pathwidth.

Theorem 3 (Kinnersley [11])**.** *The pathwidth of every graph equals its vertex separation number.*

Treewidth is the following decision problem: Given a graph G and an integer k, is the treewidth of G at most k? The problems Pathwidth and Vertex Separation Number are defined analogously.

In 1987, Arnborg, Corneil, and Proskurowski established NP-completeness of Treewidth in the class of co-bipartite graphs [1]. Ten years later, Bodlaender and Thilikos [6] proved that Treewidth is NP-complete on graphs with maximum degree at most 9. Monien and Sudborough [15] proved that Vertex Separation Number is NP-complete on planar graphs with maximum degree at most 3. Combining this result with Theorem 3 directly shows the following.

Theorem 4 (Monien and Sudborough [15])**.** Pathwidth *is NP-complete on planar graphs with maximum degree at most* 3.

A well-known type of graphs are the *walls*. A wall with r rows and c columns has $r \times c$ vertices; for an illustrating example, see Figure 1.[1]

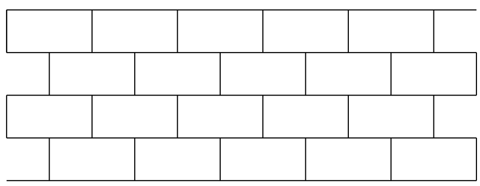

Figure 1: A wall with 5 rows and 12 columns.

[1] The most common notion of a wall does not have the vertices of degree one, which we see at the bottom left and top right corner of Figure 1. We keep these degree-one vertices, for slightly easier notation.

It is well known that the pathwidth and treewidth of an r by c grid are both equal to $\min\{r, c\}$, see, e.g. [4, Lemmas 87 and 88]. Since any wall is a subgraph of a grid, the upper bound also holds for walls, and the standard construction gives the following result.

Lemma 5 (Folklore)**.** *Let $B_{r,c}$ be a wall with r rows and c columns. Then $\mathtt{tw}(B_{r,c}) \leqslant \mathtt{pw}(B_{r,c}) \leqslant r$ and there is a path decomposition (P, β) of $B_{r,c}$ of width r with $\beta(p_1)$ the set of vertices in the first column of $B_{r,c}$, and $\beta(p_q)$ the set of vertices in the last column of $B_{r,c}$, where p_1 and p_r are the two endpoints of P.*

A *linear ordering* of a graph G is a bijection $f : V(G) \to \{1, \dots, n\}$. The *cutwidth* of a linear ordering of G is

$$\max_{i \in [n]} \Big| \big\{ \{v, w\} \in E(G) \;\big|\; f(v) \leqslant i < f(w) \big\} \Big|.$$

The *cutwidth* of a graph G, denoted by $\mathtt{cw}(G)$, is the minimum cutwidth of a linear ordering of G. Observe that if a graph H can be obtained from a graph G by deleting edges and/or contracting vertices of degree 2 to a neighbour, then $\mathtt{cw}(H) \leqslant \mathtt{cw}(G)$.

The Cutwidth problem asks to decide, for a given graph G and integer k, whether the cutwidth of G is at most k. Monien and Sudborough [15] showed that Cutwidth is NP-complete on graphs of maximum degree three (using the problem name Minimum Cut Linear Arrangement). As their proof does not generate vertices of degree one, and the cutwidth of a graph does not change by subdividing an edge, from their proof, the next result follows.

Theorem 6 (Monien and Sudborough [15])**.** Cutwidth *is NP-complete on cubic graphs.*

3 A simple proof for co-bipartite graphs

In this section, we give a new simple proof that Treewidth is NP-complete. Our proof borrows elements from the NP-completeness proof from Arnborg, Corneil and Proskurowski [1], but uses an easy transformation from Pathwidth.

Let G be a graph. We denote by $F(G)$ the graph obtained from G as follows. The vertices of $F(G)$ consist of two copies v and v' for every $v \in V(G)$; we denote by V and V' the sets $V(G)$ and $\{v' \mid v \in V(G)\}$, respectively. Moreover, the graph $F(G)$ contains for every $v \in V(G)$ an edge between v and v', and for every edge $\{u, v\} \in E(G)$, it contains one edge between u and v' and one edge between v and u'. Finally, $F(G)$ contains all edges between every pair of distinct vertices in V and every pair of distinct vertices in V'. Note that each of the sets V and V' is a clique in $F(G)$. Hence, G is co-bipartite. An example is given in Figure 2.

Lemma 7. *Let G be a graph. Then, $\mathtt{tw}(F(G)) = \mathtt{pw}(F(G)) = n + \mathtt{pw}(G)$, where $n = |V(G)|$.*

Proof. First, we show that $\mathtt{pw}(F(G)) \leqslant n + \mathtt{pw}(G)$. Let $k = \mathtt{pw}(G)$. Take a path decomposition (P, β) of G of width k, with $P = (p_1, \dots, p_r)$. Now, let $\gamma(p_i)$ be a set of vertices of $F(G)$ defined as follows:

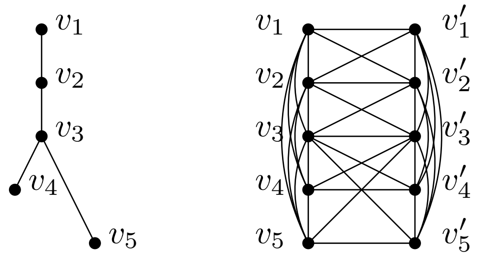

Figure 2: A graph G with $F(G)$.

- For each $v \in V(G)$ such that there is a $j \geqslant i$ with $v \in \beta(p_j)$, add v to $\gamma(p_i)$.
- For each $v \in V(G)$ such that there is a $j \leqslant i$ with $v \in \beta(p_j)$, add v' to $\gamma(p_i)$.

An example of this construction, applied to the graphs G and $F(G)$ of Figure 2, is given in Figure 3.

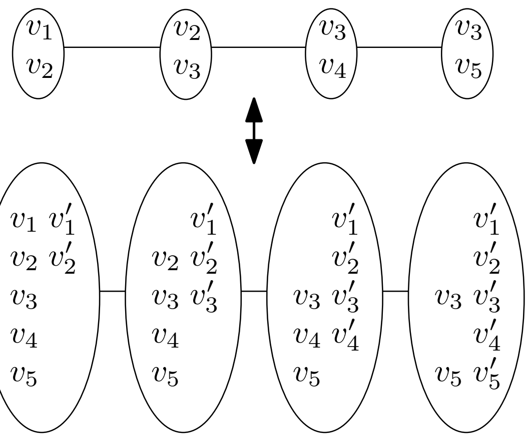

Figure 3: A path decomposition of the graph G from Figure 2 and the corresponding path decomposition of $F(G)$.

We claim that (P, γ) is a path decomposition of $F(G)$ of width $n + k$. We first verify that (P, γ) is a path decomposition. The first and third conditions of path decompositions are clearly satisfied. Notice that $V \subseteq \gamma(p_1)$, and $V' \subseteq \gamma(p_r)$. So, for each edge in $F(G)$ between two vertices in V, or between two vertices in V', there is a bag in (P, γ) containing the two endpoints of the edge, namely, the bag corresponding to the node p_1 or p_r, respectively. Consider an edge $\{v, v'\}$ for a vertex $v \in V(G)$. There is a node p_v with $v \in \beta(p_v)$, and therefore $v, v' \in \gamma(p_v)$. Consider an edge $\{v, w'\}$ in $F(G)$, corresponding to an edge $\{v, w\} \in E(G)$. There is a node p_{vw} with $v, w \in \beta(p_{vw})$. Now, $v, v', w, w' \in \gamma(p_{vw})$.

To see that the width is $n + k$, consider some bag $\gamma(p_i)$ and a vertex $v \in V(G)$. There are three possible cases:

1. For each j with $v \in \beta(p_j)$, $j > i$. Now, $v \in \gamma(p_i)$; $v' \notin \gamma(p_i)$.

2. For each j with $v \in \beta(p_j)$, $j < i$. Now, $v' \in \gamma(p_i)$; $v \notin \gamma(p_i)$.

3. If the previous two cases do not hold, then there is a $j \leqslant i$ with $v \in \beta(p_j)$, and a $j' \geqslant i$ with $v \in \beta(p_{j'})$. From the definition of path decompositions, it follows that $v \in \beta(p_i)$. From the construction of γ, we have $v, v' \in \gamma(p_i)$.

In each of the cases, we have one vertex more in $\gamma(p_i)$ than in $\beta(p_i)$, so for each node, the size of its γ-bag is exactly n larger than the size of its β-bag. The claim follows.

Now, assume that the treewidth of G equals ℓ. From Lemma 2(2), it follows that we can assume we have a path decomposition (P, γ) of $F(G)$ of width ℓ, with P having successive bags $p_1, p_2, \ldots, p_r$, and with $V \subseteq \gamma(p_1)$ and $V' \subseteq \gamma(p_2)$.

We now define a path decomposition (P, δ) of G, as follows. For each node x on P, set $\delta(x) = \{v \in V \mid v \in \gamma(x) \wedge v' \in \gamma(x)\}$. (Note that this is the reverse of the operation in the first part of the proof; compare with Figure 3.)

We now verify that (P, δ) is indeed a path decomposition of G. For each vertex v, the set $\{v, v'\}$ is an edge in $F(G)$, so there is a node x_v with $v, v' \in \gamma(x_v)$, hence $v \in \delta(x_v)$. For each edge $\{v, w\} \in E(G)$, the set $\{v, v', w, w'\}$ forms a clique in $F(G)$, so there is a node x_{vw} with $\{v, v', w, w'\} \subseteq \gamma(x_{vw})$ (see Lemma 1(1)). Hence $v, w \in \delta(x_{vw})$. Finally, for each $v \in V(G)$, the set of nodes x with $v \in \delta(x)$ is the intersection of the nodes with $v \in \gamma(x)$ and the nodes with $v' \in \gamma(x)$; the intersection of connected subtrees is connected, so the third condition in the definition of path (tree) decompositions also holds.

Finally, we show that the width of (P, δ) is $\ell - n$. Consider a vertex v, and $i \in [r]$. There must be i_v with $\{v, v'\} \subseteq \gamma(p_{i_v})$. If $i \leqslant i_v$, then $v \in \gamma(p_i)$; if $i \geqslant i_v$, then $v' \in \gamma(p_i)$ (using that $v \in \gamma(p_1)$ and $v' \in \gamma(p_r)$). So, we have $\{v, v'\} \cap \gamma(p_i) \neq \emptyset$.

Now, for each node p_i, $i \in [r]$, for each vertex v, we have that $\gamma(p_i)$ contains both vertices from the set $\{v, v'\}$ when $v \in \delta(p_i)$, and $\gamma(p_i)$ contains exactly one vertex from the set $\{v, v'\}$ when $v \notin \delta(p_i)$. So, $|\gamma(p_i)| = |\delta(p_i)| + n$. As this holds for each bag, we have that the width of (P, γ) is exactly n larger than the width of (P, δ). It follows that $\mathtt{pw}(G) \leqslant \mathtt{tw}(F(G)) - n \leqslant \mathtt{pw}(F(G)) - n$, which shows the result. □

Lemma 7, together with the NP-completeness of Vertex Separation Number [15], and the equivalence between pathwidth and vertex separation number (Theorem 3), leads to an alternative simple proof of NP-completeness of Treewidth in the class of co-bipartite graphs.

Corollary 8. Treewidth *is NP-complete on co-bipartite graphs.*

One can obtain a proof of the NP-completeness of Treewidth on graphs with maximum degree five by combining the proof above with the technique of replacing a clique with a wall or grid (as in [6] or in the next section). Instead of this, we give in the next section a proof that reduces from Cutwidth and shows NP-completeness of Treewidth on cubic graphs.

4 Cubic graphs

The construction leading to an NP-completeness proof for TREEWIDTH on cubic graphs uses a few steps, which we now summarize. The first step is a simplified version of the NP-completeness proof from Arnborg, Corneil and Proskurowski [1]; the second step follows the idea of Bodlaender and Thilikos [6] to replace the cliques by grids or walls. After this step, we have a graph with maximum degree 7. In the third step, we replace vertices of degree more than 3 by trees of maximum degree 3, and show that this step does not change the treewidth. The fourth step makes the graph 3-regular by simply contracting over vertices of degree 2. For later use, we also give bounds on the pathwidth of the constructed graphs.

Theorem 9. TREEWIDTH *is NP-complete on cubic graphs.*

Proof. We use a transformation from CUTWIDTH on 3-regular graphs.

Let G be an n-vertex 3-regular graph and k an integer. Using a sequence of intermediate steps and intermediate graphs G_1, G_2, G_3, we construct a 3-regular graph G_4 with the property that G has cutwidth at most k, if and only if G_4 has treewidth at most $3n + k + 2$.

Step 1: From Cutwidth to Treewidth The first step is a streamlined version of the proof from Arnborg, Corneil and Proskurowski [1]. For each vertex $v \in V(G)$, we take a set $A_v = \{v^1, v^2, v^3\}$, which has three copies of v.

For each edge $e \in E(G)$, we have a set $B_e = \{e^1, e^2\}$, which consists of two vertices that represent the edge.

Let $A = \bigcup_{v\in V(G)} A_v$ and $B = \bigcup_{e\in E(G)} B_e$. We create G_1 by taking $A \cup B$ as vertex set, turning A into a clique, turning B into a clique, and for each pair v, e with v an endpoint of e, adding edges between all vertices in A_v and all vertices in B_e.

Claim 10. *Let G and G_1 be as above. Then,* $\texttt{tw}(G_1) = \texttt{pw}(G_1) = \texttt{cw}(G) + 3n + 2$.

Proof. First, assume G has cutwidth k, and let f be a linear ordering of G of cutwidth k, and denote the ith vertex in the linear ordering as $v_i = f^{-1}(i)$.

Build a path decomposition (P, β) with P the path with nodes $p_1, \ldots, p_n$. For $i \in [n]$, set

$$\begin{aligned}\beta(p_i) = &\left\{ v_j^a \;\middle|\; j \geqslant i \wedge a \in \{1, 2, 3\} \right\} \\ &\cup \left\{ e^b \;\middle|\; e = \{v_j, v_{j'}\} \in E(G) \wedge \min\{j, j'\} \leqslant i \wedge b \in [2] \right\}.\end{aligned}$$

That is, we take the representatives of the vertices $v_i, v_{i+1}, \ldots, v_n$, and all vertices that represent an edge with at least one endpoint in $\{v_1, v_2, \ldots, v_i\}$.

We can verify that (P, β) is a path decomposition of G_1. From the construction, it directly follows that $A \subseteq \beta(p_1)$ and $B \subseteq \beta(p_n)$. For the second condition of path decompositions, it remains to look at edges in G_1 with one vertex of the form v_i^a and one vertex of the form e^b. Necessarily, v_i is an endpoint of e, and now we can note that both vertices are in bag $\beta(p_i)$. From the construction, it directly follows that the third condition of path decompositions is fulfilled.

To show that the width of this path decomposition is at most $k+3n+2$, we use an accounting system. Consider $\beta(p_i)$. Give each vertex $v \in V(G)$ three credits, except v_i, which gets six credits. Each edge that 'crosses the cut', i.e. it belongs to the set $\{\{v,w\} \in E(G) \mid f(v) \leqslant i < f(w)\}$, gets one credit. All other edges get no credit. We handed out at most $k+3n+3$ credits. We now redistribute these credits to the vertices in $\beta(p_i)$. Each vertex v_j, $j \geqslant i$, gives one credit to each vertex of the form v_j^a, $a \in \{1,2,3\}$. For an edge $e = \{v_j, v_{j'}\}$, with $j < i$ and $j' < i$, the vertices e^1 and e^2 get, respectively, a credit from v_j and $v_{j'}$. For an edge $e = \{v_j, v_{j'}\}$, with $j \leqslant i < j'$, the vertices e^1 and e^2 get, respectively, a credit from v_j and a credit from e. Now, each vertex and edge precisely spends its credit: a vertex v_j with $j < i$ gives one credit to each of its incident edges, v_i gives one credit to each of its copies v_i^1, v_i^2, v_i^3, and one credit to each of its incident edges, and v_j with $j > i$ gives one credit to each of its copies v_j^1, v_j^2, v_j^3. Each vertex in the bag $\beta(p_i)$ gets one credit, so the size of the bag is at most $k+3n+3$. As this holds for each bag, the width of the path decomposition is at most $k+3n+2$.

Now, assume that we have a tree decomposition (T,γ) of G_1 of width ℓ. By Lemma 1(1), as A and B are cliques, there is a bag p_1 with $A \subseteq \gamma(p_1)$, and a bag p_r with $B \subseteq \gamma(p_r)$. As in the proof of Lemma 2, we can remove all bags not on the path from p_1 and p_r, and still keep a tree decomposition of G_1. So, we can assume we have a path decomposition (P,γ) of width at most ℓ of G_1, where P is a path with successive vertices $p_1, p_2, \ldots, p_r$, and $\gamma(p_1) = A$ and $\gamma(p_r) = B$.

For each $v \in V(G)$, set $g(v)$ to the maximum i such that $\{v^1, v^2, v^3\} \subseteq \beta(p_i)$. (As $\{v^1, v^2, v^3\} \subseteq A \subseteq \beta(p_1)$, $g(v)$ is well defined and in $[r]$.)

Take a linear ordering f of G such that for all $v, w \in V(G)$, $g(v) < g(w) \Rightarrow f(v) < f(w)$. (That is, order the vertices with respect to increasing values of g, and arbitrarily break the ties when vertices have the same value $g(v)$.) We claim that f has cutwidth at most $\ell - 3n - 2$.

Consider a vertex $v \in V(G)$, and assume $g(v) = i'$. Let e be an edge incident to v. The set $\{v^1, v^2, v^3, e^1, e^2\}$ is a clique in G_1, so there is an i_e with $\{v^1, v^2, v^3, e^1, e^2\} \subseteq \beta(p_{i_e})$. From the definition of path decompositions and the construction of g, we have $i_e \leqslant i'$. As $\{e^1, e^2\} \subseteq \beta(p_{i_e}) \cap \beta(p_r)$, we have that $\{e^1, e^2\} \subseteq \beta(p_{i'})$.

Now, consider an $i \in [n]$. Let $v = f^{-1}(i)$ be the ith vertex of the ordering and $C = f^{-1}[i]$ be the first i vertices in the linear ordering. Let E^1 be the set of edges with exactly one endpoint in C, and let E^2 be the set of edges with both endpoints in C. Let $i' = g(v)$. We now examine which vertices belong to $\beta(p_{i'})$:

- By definition, v^1, v^2, v^3.

- For each $w \in V(G) \setminus C$, there is an $i_w \geqslant i'$ with $\{w^1, w^2, w^3\} \subseteq \beta(p_{i_w})$, hence w^1, w^2, and w^3 are in $\beta(p_{i'})$. (We use here that these vertices are in $\beta(p_1)$.) The number of such vertices is $3n - 3i$.

- For each edge $e \in E^1 \cup E^2$, from the discussion above it follows that there is an $i_e \leqslant i'$ with $e^1, e^2 \in \beta(p_{i_e})$, and, as these vertices are in $\beta(p_r)$, we have $\{e^1, e^2\} \subseteq \beta(p_{i'})$.

Thus, the size of $\beta(p_{i'})$ is at least $3n - 3i + 3 + 2 \cdot |E_1| + 2 \cdot |E_2|$. As each vertex in C is incident to exactly three edges, we have $3i = |E_1| + 2 \cdot |E_2|$. Now, $\ell \geqslant |\beta(p_{i'})| - 1 \geqslant 3n - 3i + 2 + 2 \cdot |E_1| + 2 \cdot |E_2| = 3n + 2 + |E_1|$. It follows that the size of the cut can be bounded as follows:

$$\Big|\big\{\{x, y\} \in E(G) \;\big|\; f(x) \leqslant i < f(y)\big\}\Big| = |E_1| \leqslant \ell - 3n - 2\,.$$

As this holds for each $i \in [n]$, the bound of $\ell - 3n - 2$ on the cutwidth of f follows.

We have thus shown that $\mathtt{pw}(G_1) \leqslant \mathtt{cw}(G) + 3n + 2$ and that $\mathtt{cw}(G_1) \leqslant \mathtt{tw}(G_1) - 3n - 2$. Together with the inequality $\mathtt{tw}(G_1) \leqslant \mathtt{pw}(G_1)$, this proves the claim. □

Step 2: The wall construction In the second step, we use a technique from Bodlaender and Thilikos [6]. We construct a graph G_2 from the graph G_1 by removing the edges between vertices in A and the edges between vertices in B; then, we add a wall with $3n$ rows and $24n$ columns, and add a matching from the vertices in the last column of the wall to the vertices in A. Similarly, we add another wall with $3n$ rows and $24n$ columns, and add a matching from the vertices in the first column of this wall to the vertices in B.

As applying the wall construction to a graph obtained from the first step would be unwieldy, the example in Figure 4 shows the wall construction applied to the graph from the previous section.

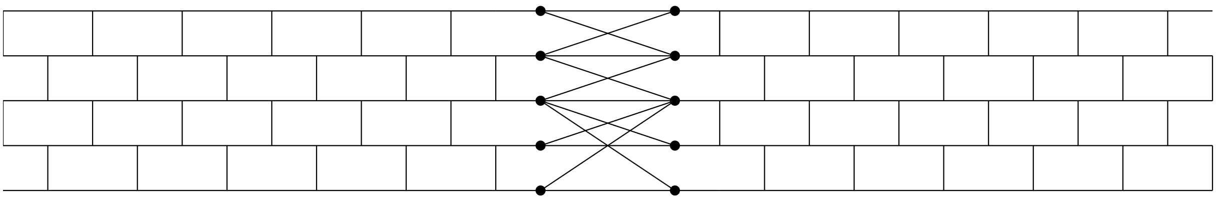

Figure 4: Illustration of the wall construction. Here, it is applied to the graphs from Figure 2, and the number of columns shown is smaller than that in the actual construction.

Claim 11. $\mathtt{tw}(G_1) = \mathtt{pw}(G_1) = \mathtt{tw}(G_2) = \mathtt{pw}(G_2)$. *Moreover, there is a path decomposition of* G_2 *of optimal width with a node* x_A *with* $A \subseteq \beta(x_A)$ *and a node* x_B *with* $B \subseteq \beta(x_B)$.

Proof. Suppose we have a tree decomposition (T, β) of G_2 of optimal width k. By Lemma 1(3), there is a node x such that each connected component of $G_2 \setminus \beta(x)$ contains at most $36n^2$ vertices of the left wall.

We claim that $\beta(x)$ must contain a vertex of each row from the left wall. Suppose not. Let W_r be the vertex set of a row that does not contain a vertex from $\beta(x)$. Each column contains a vertex from W_r, and each pair of two successive columns induces a connected subgraph. As we have $24n$ columns in the left wall, there are at least $12n - |\beta(x)|$ pairs of successive columns that do not contain a vertex from $\beta(x)$. All vertices in such a pair of columns belong to the same connected component of $G_2 \setminus \beta(x)$ as W_r: the two columns are connected, contain vertices from W_r, but no vertex from $\beta(x)$. This gives $12n - |\beta(x)|$ disjoint sets of $6n$ vertices in this connected component. And, thus, $G_2 \setminus \beta(x)$ has a connected component with at least $(6n) \cdot (12n - \beta(x)) \geqslant 72n^2 - 6n \cdot (k+1) > 36n^2$, using that $k \leqslant |E(G)| = 3n/2$. This contradicts the stated property of x.

By Lemma 1(2), (T, β) is also a tree decomposition of the graph obtained from G_2 by adding edges between each pair of vertices in $\beta(x)$. Apply the same step to the right wall. We see that (T, β) is a tree decomposition of width k of a graph that for each pair of rows in the left wall contains an edge between a pair of vertices from these rows, and similarly for the right wall. Call this graph G_2'.

The rightmost vertex of a row of the left wall is adjacent to one vertex in A; we now contract each row of the left wall to this vertex in A. Similarly, we contract each row of the right wall to the vertex in B that is adjacent to the leftmost vertex of the row. Observe now that the graph obtained by these contractions is G_1. Furthermore, G_1 is a minor of G_2', and thus, by Lemma 1(5), $\mathtt{tw}(G_1) \leqslant \mathtt{tw}(G_2') \leqslant k$.

By Lemma 2, $\mathtt{tw}(G_1) = \mathtt{pw}(G_1)$, and there is a path decomposition (P, γ) of G_1 of optimal width ℓ such that $A \subseteq \gamma(p_1)$ and $B \subseteq \gamma(p_q)$, where p_1 and p_q are the endpoints of P.

We can now build a path decomposition of G_2 of the same width ℓ as follows: first, take the successive bags of a path decomposition of the left wall, of width $3n$, where we can end with a bag that contains all vertices of A. Then, we take the bags of (P, γ). Now, we add a path decomposition of the right wall, of width $3n$, that starts with a bag containing all vertices in B. □

Step 3: Making the graph subcubic Let us observe that the maximum degree of a vertex in G_2 is seven. Indeed, any vertex in A has one neighbour in the wall, and six neighbours in B (the vertex it represents has three incident edges, and each is represented by two vertices). Similarly, any vertex in B has degree seven: again, one neighbour in the wall, and six neighbours in A (each endpoint of the edge it represents is represented by three vertices). Vertices in the walls have degree at most three.

Given G_2, we build a subcubic graph G_3. We do this by replacing each vertex in A and in B by a tree, and replacing edges to vertices in A and B by edges to leaves or the root of these trees.

For vertices v^α in A (with $v \in V(G)$, $\alpha \in [3]$), we take an arbitrary tree with a root of degree 2, all other internal vertices of degree 3, and six leaves. The root (which we denote by the name of the original vertex v^α) is made adjacent to the neighbour of v^α in the wall.

Each vertex $e^\alpha \in B$ (with $e \in E(G)$, $\alpha \in [2]$) is also replaced by a tree with a root of degree 2, all other internal vertices of degree 3, and six leaves, but here we need to use a specific shape of the tree. Suppose e has endpoints v and w. Figure 5 shows this tree. In particular, note that the root is made adjacent to the neighbour of e^α in the wall, the leaves that go to the subtrees that represent v are grouped together, and the leaves that go to the subtrees that represent w are grouped together.

Each edge between a vertex v^α in A and a vertex $e^{\alpha'}$ in B now becomes an edge from

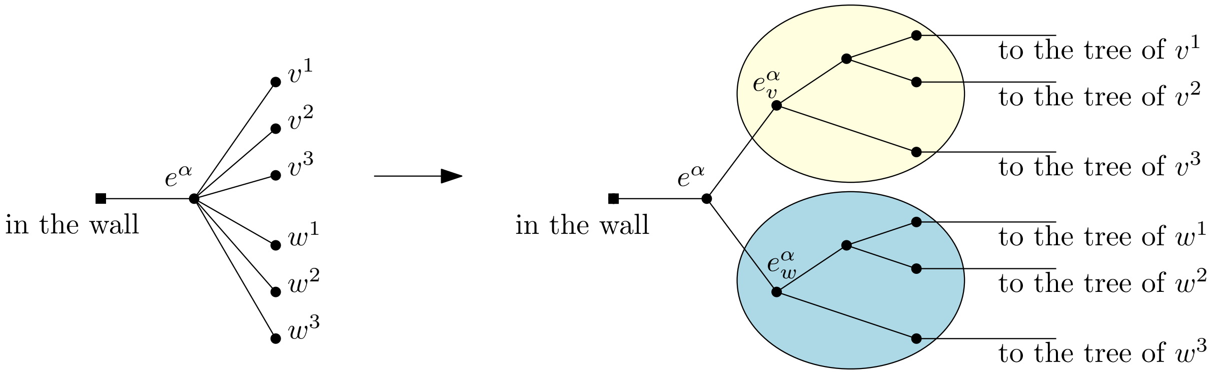

Figure 5: Replacing a vertex e^α from B by a tree; e is here the edge $\{v, w\}$.

a leaf of the tree representing v^α, to a leaf of the tree representing $e^{\alpha'}$; $\alpha \in [3]$, $\alpha' \in [2]$. The roots of the trees are made adjacent to a vertex in the wall; this is the same vertex as the wall neighbour of the original vertex in G_2.

Claim 12. *Suppose* $\mathtt{tw}(G_2) \geqslant 68$. *Then* $\mathtt{tw}(G_2) = \mathtt{pw}(G_2) = \mathtt{tw}(G_3)$. *Furthermore,* $\mathtt{pw}(G_3) \leqslant \mathtt{tw}(G_3) + 69$.

Proof. We have already established that $\mathtt{tw}(G_2) = \mathtt{pw}(G_2)$.

First, note that G_2 is a minor of G_3: we obtain G_2 from G_3 by contracting each of the new trees to its original vertex. By Lemma 1(5), we have $\mathtt{tw}(G_2) \leqslant \mathtt{tw}(G_3)$.

Suppose we have a path decomposition (P, β) of G_2 of optimal width $\ell = \mathtt{pw}(G_2) = \mathtt{tw}(G_2)$. By Claim 11, we may assume that there is a bag that contains all vertices in A, and that there is a bag that contains all vertices in B.

For each vertex $v \in V(G)$, we claim that there is a node p_{i_v} with $v^1, v^2, v^3 \in \beta(p_{i_v})$ and $e^1, e^2 \in \beta(p_{i_v})$ for each of the three edges e incident to v. It is possible to derive this from the proofs of Claims 10 and 11; a more compact argument is the following: The pair (P, β) is also a path decomposition of the graph $G + \mathit{clique}(A) + \mathit{clique}(B)$, obtained from G_2 by adding edges between each pair of vertices in A, and each pair of vertices in B (since there is a bag containing all vertices of A and a bag containing all vertices of B and by Lemma 1(2)). The claim now follows from Lemma 1(1) by observing that these nine vertices (vertices v^1, v^2, v^3, and vertices e^1, e^2 for each edge e incident to v) form a clique in $G + \mathit{clique}(A) + \mathit{clique}(B)$.

Now, we can construct a tree decomposition of G_3 as follows. Take (P, β). Each vertex $v \in A \cup B$ has a subtree in G_3 that represents v; in each bag that contains v, we replace v by the root of that subtree. For each vertex $v \in V(G)$, we add one additional bag to the tree decomposition; this bag becomes a leaf of the tree decomposition. (Note that after this step, we no longer have a path decomposition.)

Consider a vertex $v \in V(G)$. Take a new node x_v, and make x_v adjacent to p_{i_v} in the tree. Let the bag of x_v contain the following vertices: all vertices in the subtrees that represent v^1, v^2, v^3, for each edge e with v as endpoint the vertices e^1, e^1_v, e^2, e^2_v, and the descendants of e_v^1 and e_v^2 in the respective subtrees (the vertices in the yellow area in Figure 5, assuming that $e = \{v, w\}$).

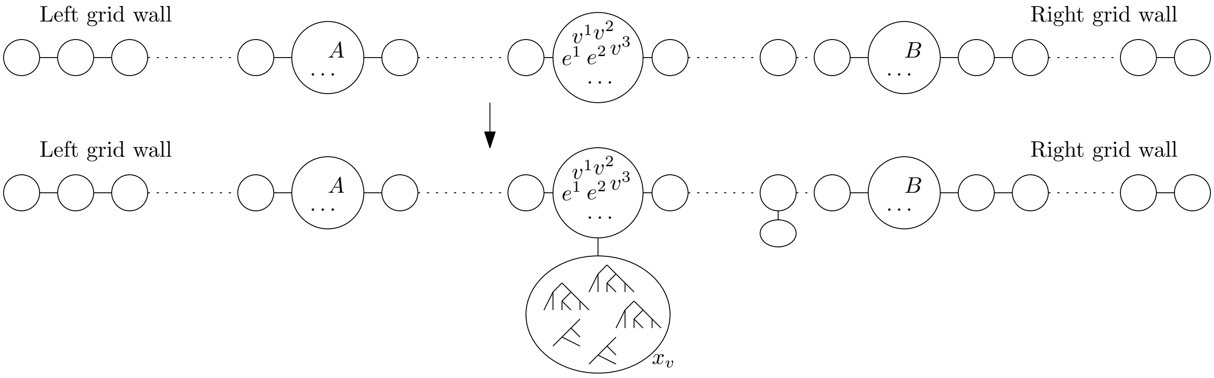

Figure 6: Illustration of the proof. The decomposition before and after adding the new node x_v.

Each vertex in A is represented by a binary tree with a root of degree two and six leaves, so by eleven vertices. For each of the three edges incident to v, we have two subtrees of which we take six vertices each, so the total size of this new bag is $3 \cdot 11 + 3 \cdot 2 \cdot 6 = 69$. One easily verifies that we have a tree decomposition of G_3, and as the original bags keep the same size, when $\ell \geqslant 68$ we have a tree decomposition of G of width at most ℓ.

By merging the new bags with their parents, we obtain a path decomposition of G_3 of width $\ell + 69$, proving the second part of the claim. □

Claim 10 and Claim 11 imply that, by taking a sufficiently large n (e.g. $n \geqslant 22$ works), we can assume that $\mathtt{tw}(G_2) \geqslant 68$.

Step 4: Making the graph 3-regular The fourth step is simple. Note that when the treewidth of a graph is at least three, the treewidth does not change when we contract a vertex of degree at most two to a neighbour (see [2]), possibly removing parallel edges. Note that this also cannot increase the pathwidth of the graph. We apply this step as many times as possible, and let G_4 be the resulting graph. The graph G_4 is a 3-regular graph, and, when $n \geqslant 22$, its treewidth equals the treewidth of G_1, which is $\mathtt{cw}(G)+3n+2$. As the pathwidth is not increased, by Claim 12 we have:

Claim 13. *$\mathtt{pw}(G_4) \leqslant \mathtt{tw}(G_4) + 69$.*

As we can construct G_4 in polynomial time, this completes the transformation, and we can conclude that TREEWIDTH is NP-complete on 3-regular graphs. □

5 Special cases

In this section, we give three NP-completeness proofs for TREEWIDTH on special graph classes, which follow from minor modifications of the proof of Theorem 9.

5.1 Regular graphs

We first observe that for any fixed $d \geqslant 4$, Treewidth is NP-complete on d-regular graphs.

Proposition 14. *For each $d \geqslant 3$,* Treewidth *is NP-complete on d-regular graphs.*

Proof. The result for $d = 3$ was given as Theorem 9.

A small modification of the proof of Theorem 9 gives the result for 4-*regular* graphs: instead of using a wall, use a grid. At the borders of this grid, we have vertices of degree less than 3. We can avoid these by first contracting vertices of degree 2, and then noting that there is a perfect matching with the vertices of degree 3 at the sides of the grid. Replace each edge in this matching by a small subgraph, as shown in Figure 7. Note that this step increases the degree of v and w by one, while, when the treewidth of G is at least 5, the step will not change the treewidth of the graph.

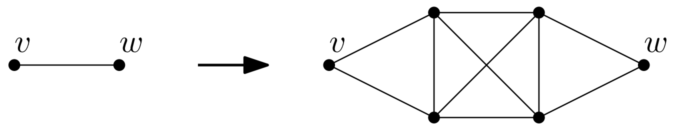

Figure 7: Increasing the degree of two adjacent vertices by one.

In the step where we change vertices of degree 7 to vertices of degree 3 by replacing a vertex by a small tree, we instead use a tree with the root having two children, each with three children. These roots are made adjacent to the grid. Now, the roots have degree 3, and we add an arbitrary perfect matching between these root vertices in A, and similarly for B. (Note that in the construction, there is a bag containing all roots for A, and similarly B; these sets have even size.) This gives the result for $d = 4$.

Consider the following gadget. Take a clique with $d+1$ vertices, and remove one edge, say $\{x, y\}$, from this clique. For a vertex v in a graph G, add an edge from x to v, and an edge from y to v. See Figure 8.

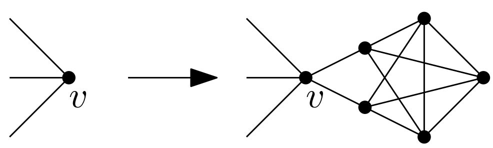

Figure 8: Increasing the degree of a vertex: if $\mathtt{tw}(G) \geqslant 4$, then the step increases the degree of v from 3 to 5, but does not change the treewidth.

If G has treewidth at least d, then this step increases the degree of v by 2 without changing the treewidth. Now, if d is odd, we can take an instance of the hardness proof on 3-regular graphs, and add to each vertex of that instance $(d-3)/2$ copies of this gadget. We obtain an equivalent instance that is d-regular. If d is even, we add $(d-4)/2$ copies of the gadget to an instance of the hardness proof on 4-regular graphs. □

5.2 Grid graphs

A *d-dimensional grid graph* is a finite induced subgraph of the infinite d-dimensional grid. Observe that d-dimensional grid graphs have degree at most $2d$, and in particular the 3-dimensional grid graphs have degree at most 6. As a consequence of lowering the degree of hard TREEWIDTH instances from 9 to at most 6, we can show that computing the treewidth of 3-dimensional grid graphs is NP-complete. Since we lowered the degree of hard instances down to at most 3, we can even show the following.

Proposition 15. TREEWIDTH *is NP-complete on subcubic 3-dimensional grid graphs.*

Proof. The argument is simply that every n-vertex (sub)cubic graph admits a subdivision of polynomial size that is a 3-dimensional grid graph. We give a simple such embedding.

We reduce from TREEWIDTH on cubic graphs, which is NP-hard by Theorem 9. Let G be any cubic graph, $v_0, v_1, \ldots, v_{n-1}$ its vertices, and $e_1, e_2, \ldots, e_{3n/2}$ its edges. We build a subcubic induced subgraph H of the $(6n-1) \times (3n+1) \times 3$ grid that is a subdivision of G. In particular, $\texttt{tw}(H) = \texttt{tw}(G)$ and H has $\mathcal{O}(n^2)$ vertices and edges, implying the desired conclusion.

For each $i \in [0, n-1]$, vertex v_i is encoded by the path made by the 5 vertices $(x, 0, 0)$ with $x \in [6i, 6i+4]$. We arbitrarily assign $(6i, 0, 0)$, $(6i+2, 0, 0)$, $(6i+4, 0, 0)$ each with a distinct neighbour of v_i in G, say $v_{i(0)}$, $v_{i(1)}$, $v_{i(2)}$, respectively.

Every edge $e_k = \{v_i, v_j\}$ of G with $i < j$ is encoded in the following way. Let $a, b \in [0, 2]$ be such that $i(a) = j$ and $j(b) = i$. We build a path from $(6i+2a, 0, 0)$ to $(6j+2b, 0, 0)$ with degree-2 vertices, by first adding all the vertices $(6i+2a, y, 0)$ and $(6j+2b, y, 0)$ for $y \in [2k]$, then bridging $(6i+2a, 2k, 0)$ and $(6j+2b, 2k, 0)$ by adding $(6i+2a, 2k, 1)$, $(6i+2a, 2k, 2)$, $(6i+2a+1, 2k, 2)$, $(6i+2a+2, 2k, 2), \ldots, (6j+2b-1, 2k, 2)$, $(6j+2b, 2k, 2)$, and $(6j+2b, 2k, 1)$. An example of this construction is illustrated in Figure 9.

This finishes the construction of H. All of its vertices have degree 2, except the vertices of the form $(6i+2, 0, 0)$, which have degree 3. It is easy to see that H is a subdivision of G (where each edge gets subdivided at most $12n+5$ times). □

We can easily adapt the previous proof to show hardness for finite subcubic (non-induced) subgraphs of the $\infty \times \infty \times 2$ grid.

5.3 Cubic line graphs

In this section, we combine some simple observations, earlier proofs from this paper, and the NP-hardness proofs of Monien and Sudbourough [15], to obtain that TREEWIDTH is NP-hard for cubic line graphs.

For a graph G, the line graph $L(G)$ of G is obtained by taking $E(G)$ as the set of vertices, and adding an edge between two distinct $e, e' \in E$ if and only if e and e' share an endpoint. Consider a cubic graph G. Now, subdivide each edge in G once, and then take the line graph H of the latter graph. In other words, we can obtain H from G by replacing each vertex v of G by a triangle and each edge of G by an edge between corresponding triangles, using each vertex in a triangle once for an external edge; see Figure 10 and Figure 11. Note that H is again a cubic graph.

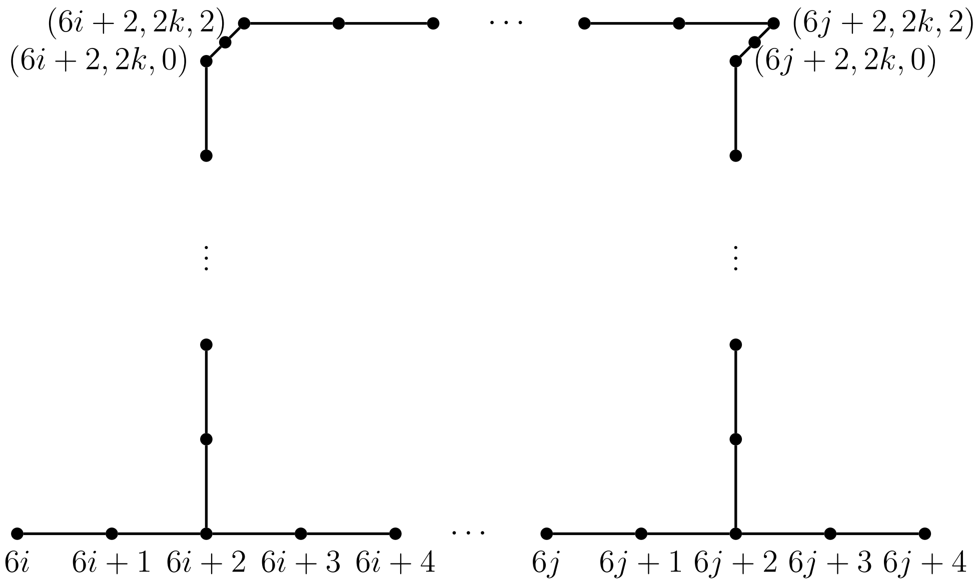

Figure 9: Embedding of a cubic graph G into a subcubic 3-dimensional grid graph employed in the proof of Proposition 15. The figure illustrates the two paths representing the vertices v_i and v_j in G together with the path corresponding to the edge e_k between v_i and v_j in G. Note that $i(1) = j$ and $j(1) = i$.

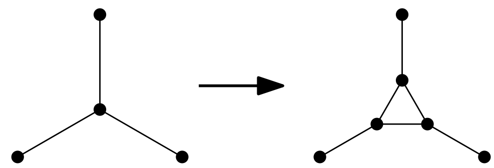

Figure 10: A local transformation that corresponds to taking the line graph of a subdivision of a cubic graph.

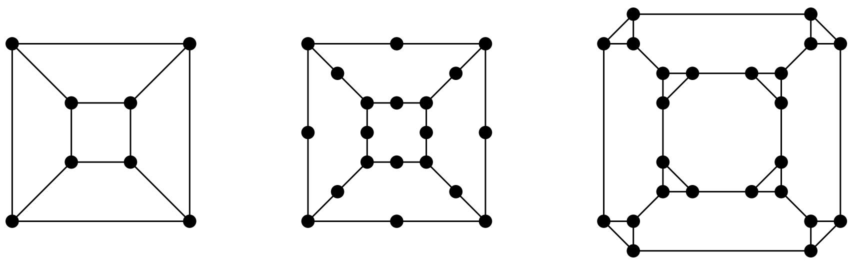

Figure 11: A graph, its subdivision, and the line graph of the subdivision.

We start with a simple observation. It is interesting to note that taking the line graph of the subdivision of a cubic graph increases the pathwidth only by a small additive term, whereas in many other cases, taking the line graph can increase of the pathwidth or treewidth by a multiplicative factor; see the study by Harvey and Wood [9].

Lemma 16. *Let G be a cubic graph and let H be the line graph of the graph obtained by subdividing each edge of G once. Then* $\texttt{cw}(G) \leqslant \texttt{cw}(H) \leqslant \texttt{cw}(G) + 2$ *and* $\texttt{pw}(G) \leqslant \texttt{pw}(H) \leqslant \texttt{pw}(G) + 4$.

Proof. Note that contracting each triangle in H representing the edges incident to a degree-3 vertex of G results in the graph G. Hence, G is a minor of H and thus, by Lemma 1(5), the pathwidth of G is at most the pathwidth of H. Also, G can be obtained from H by edge deletions (remove from each triangle one edge) and contractions over vertices of degree 2, so the cutwidth of G is at most the cutwidth of H.

Assume that we have a linear ordering f of G of cutwidth k. Replace each vertex v of G by the three vertices in H that represent its incident edges, in the order of the other endpoints of these edges. This gives a linear ordering of H of cutwidth at most $k+2$. This can be seen by a simple case analysis. If w immediately follows v in f, then the same number of edges cross the cut between v and w as well as the cut between the last vertex of the triangle of v and the first vertex of the triangle of w. Each cut between vertices in a triangle representing v has size at most $k+2$: two edges from the triangle, and at most k other edges, as these other edges are a subset of one of the cuts left or right from v in f.

For each graph of maximum degree three, we have that the cutwidth equals a parameter called search number (see [14, Corollary 3.3]), and for each graph G, the pathwidth of G is at most the search number of G, which is at most $\texttt{pw}(G) + 2$ (see [8, Theorem 2.1]); the result now follows. □

Corollary 17. *Let G be a cubic graph with* $\texttt{pw}(G) \leqslant \texttt{tw}(G) + 69$*, and let H be the line graph of the graph obtained by subdividing each edge of G once. Then* $\texttt{tw}(G) \leqslant \texttt{tw}(H) \leqslant \texttt{tw}(G) + 73$.

Proof. As G is a minor of H (contract the edges of each triangle), we have $\texttt{tw}(G) \leqslant \texttt{tw}(H)$ by Lemma 1(5). Using Lemma 16, we have $\texttt{tw}(H) \leqslant \texttt{pw}(H) \leqslant \texttt{pw}(G)+4 \leqslant \texttt{tw}(G)+69+4 = \texttt{tw}(G) + 73$. □

The main idea of the remainder of the proof in this section is the following. If we can show that for cubic graphs with pathwidth at most the treewidth plus 69, approximating treewidth is NP-hard for an additive term of 73, then this fact together with Corollary 17 will imply that computing the treewidth of cubic line graphs is NP-hard. Such a non-approximability result can be obtained by looking at a chain of reductions starting with Edge-Weighted Cutwidth for trees to Treewidth for cubic graphs, from [15], and our paper. For the first problem, it is easy to observe that approximating within a constant additive term is NP-hard; then, for each reduction in the chain, one can observe that the NP-hardness of approximation with a constant additive term is preserved. Finally, we observe that our construction in Section 4 indeed creates graphs with pathwidth at most treewidth plus 69. We make this proof idea formal below.

Monien and Sudborough [15] consider an edge-weighted version of Cutwidth. Here, each edge has a positive integer weight. The weighted cutwidth of a linear ordering is the maximum over all cuts of the total weight of the edges crossing the cut (whereas the standard cutwidth counts the number of edges.) Monien and Sudborough [15] show that EDGE-WEIGHTED CUTWIDTH is NP-complete for trees. Then, they give a transformation from EDGE-WEIGHTED CUTWIDTH for trees to CUTWIDTH for planar graphs. (This transformation replaces an edge of weight α by α parallel edges, and then subdivides each of these edges once.) After that, they give an elaborate transformation from CUTWIDTH for planar graphs to CUTWIDTH for cubic planar graphs.[2]

In order to keep the discussion sufficiently compact, we do not give a self-contained proof, but will instead refer to some details from proofs by Monien and Sudborough [15].

Proposition 18. *Let $c \geqslant 73$ be a positive integer.*

- *Let G_0 be an edge-weighted tree, and $k_0 > 2c$ an integer.*
- *Let G_1 be obtained by multiplying all edge weights in G_0 by $c+1$. Let $k_1 = k_0 \cdot (c+1)$.*
- *Let (G_2, k_2) be obtained from (G_1, k_1) by applying the transformation given in [15] from* EDGE-WEIGHTED CUTWIDTH *for trees to* CUTWIDTH *for planar graphs.*
- *Let (G_3, k_3) be obtained from (G_2, k_2) by applying the transformation given in [15] from* CUTWIDTH *for planar graphs to* CUTWIDTH *for cubic planar graphs.*
- *Let (G_4, k_4) be obtained from (G_3, k_3) by applying the transformation given in Section 4 from* CUTWIDTH *for cubic planar graphs to* TREEWIDTH *for cubic graphs.*
- *Let H be obtained from G_4 by subdividing each edge once and then taking the line graph.*

Then the following are equivalent.

1. *G_0 has a linear ordering with weighted cutwidth at most k_0.*
2. *G_1 has a linear ordering with weighted cutwidth at most k_1.*
3. *G_1 has a linear ordering with weighted cutwidth at most $k_1 + c$.*
4. *G_2 has cutwidth at most k_2.*
5. *G_2 has cutwidth at most $k_2 + c$.*
6. *G_3 has cutwidth at most k_3.*
7. *G_3 has cutwidth at most $k_3 + c$.*
8. *G_4 has treewidth at most k_4.*

[2] More precisely, the construction by Monien and Sudborough [15] creates a graph of maximum degree 3. This graph has no vertices of degree at most one, but some vertices of degree 2. However, when we contract each vertex of degree 2 with one of its neighbour, the construction still gives a simple graph, with the same cutwidth. So, we may assume the resulting graph is cubic.

9. G_4 *has treewidth at most* $k_4 + c$*.*

10. H *has treewidth at most* $k_4 + c$*.*

Proof. The equivalence of (1) and (2) is trivial, as we multiply all values by $c + 1$. As all weights in G_1 are a multiple of $c + 1$, each linear ordering of G_1 has weighted cutwidth which is a multiple of $c + 1$, and thus the equivalence of (2) and (3) follows.

The equivalences of (2), (4), (6), and (8) follow from the correctness of the various transformations, i.e., the proofs from [15] and Section 4.

To see the equivalences of (3), (5), (7), and (9) we have to look into the details of the respective transformations (from [15] and Section 4). Following these proofs, one can without much effort observe that the transformation indeed also preserves the equivalences for width values that are a small constant larger than k_i.

By Claim 13, G_4 satisfies $\mathtt{pw}(G_4) \leqslant \mathtt{tw}(G_4) + 69$. Hence, by Corollary 17, we have that if $\mathtt{tw}(G_4) \leqslant k_4$, then $\mathtt{tw}(H) \leqslant k_4 + 73$, so (8) implies (10), and that if $\mathtt{tw}(H) \leqslant k_4 + c$ then $\mathtt{tw}(G_4) \leqslant k_4 + c$, so (10) implies (9). The proposition now follows. □

Corollary 19. Treewidth *is NP-complete for cubic line graphs.*

Proof. Clearly, the problem is in NP. NP-hardness follows by transforming from Edge-Weighted Cutwidth for trees; the transformation first multiplies all weights by 74, and then successively carries out the transformations from [15] from Edge-Weighted Cutwidth for trees to Cutwidth for planar graphs, then to Cutwidth for cubic planar graphs, then the transformation from Section 4 to Treewidth for cubic graphs. Finally, we subdivide each edge, take the line graph, and add 4 to the parameter. Correctness follows from Proposition 18. □

We can also observe the following result.

Corollary 20. Pathwidth *is NP-complete for cubic line graphs.*

Proof. In a similar way as above, we build upon reductions given by Monien and Sudborough [15]. Start with an input of Edge-Weighted Cutwidth for trees. Multiply all weights by 5. Then, follow the reductions from [15] to Vertex Separation Number for graphs of degree at most three. Recall that the vertex separation number of a graph equals its pathwidth [11]. Note that their construction constructs graphs with vertices of degree two and three. Suppose these constructions build a graph G. Let H be the graph obtained by subdividing each edge in G between vertices of degree three, and then taking the line graph. As in Lemma 16, $\mathtt{pw}(G) \leqslant \mathtt{pw}(H) \leqslant \mathtt{pw}(G) + 4$. With a series of similar arguments to Lemma 18, we have that an instance of Edge-Weighted Cutwidth for trees is positive, if and only if the pathwidth of G is at most k, if and only if the pathwidth of H is at most $k + 4$. NP-hardness follows. □

6 Conclusions

In this paper, we gave a number of NP-completeness proofs for TREEWIDTH. The first proof is an elementary reduction from PATHWIDTH to TREEWIDTH on co-bipartite graphs; while the hardness result is long known, our new proof has the advantage of being very simple, and presentable in a matter of minutes. Our second main result is the NP-completeness proof for TREEWIDTH on cubic graphs, which improves upon the over 25-years-old bound of degree 9.

We end this paper with a few open problems. A long standing open problem is the complexity of TREEWIDTH on planar graphs. It is interesting to recall that the famous ratcatcher algorithm by Seymour and Thomas[18] solves the BRANCHWIDTH problem on planar graphs in polynomial time, but several other related questions still have the complexity status open. We list a number of open problems:

- Is TREEWIDTH for planar graphs NP-complete? Or, is there a polynomial time algorithm?
- Is there a constant c such that TREEWIDTH on graphs with genus at most c is NP-complete?
- Is there a fixed (non-planar) graph H such that TREEWIDTH is NP-complete for graphs that do not have H as a minor? (Note that for a planar graph H, there is a linear time algorithm for TREEWIDTH of graphs without H as minor, as graphs that avoid a planar graph as minor have bounded treewidth, so we can combine [17, 3].)
- Is there a (non-planar) surface S such that BRANCHWIDTH for graphs that can be drawn without crossings on S is polynomial time solvable?
- Is there a constant c such that BRANCHWIDTH on graphs with genus at most c is NP-complete?
- Is there a constant c such that BRANCHWIDTH on graphs with degree at most c is NP-complete? What is the complexity of BRANCHWIDTH of cubic graphs? (We conjecture it is NP-complete.)
- Current proofs for lower bounds for the approximation of treewidth create graphs of large minimum degree [7, 19]. Can we give improved upper or lower bound for the approximation ratio of polynomial time approximation algorithms for treewidth?

The reductions in our hardness proofs increase the parameter by a term linear in n, so shed no light on the parameterised complexity of TREEWIDTH. Hence, it would be interesting to obtain parameterised reductions (i.e. reductions that change k to a value bounded by a function of k), and also aim at lower bounds (e.g. based on the (S)ETH) on the parameterised complexity of TREEWIDTH. Very recently, Bonnet [7] showed that under the ETH, TREEWIDTH requires $2^{\Omega(n)}$ time, which also implies a lower bound of

$2^{\Omega(k)}n^{\mathcal{O}(1)}$ for Treewidth. Korhonen and Lokshtanov [12] gave an algorithm with running time $\mathcal{O}(2^{\mathcal{O}(k^2)}n^{\mathcal{O}(1)})$; thus, as an open problem we have to find reductions that increase the $2^{\Omega(k)}$ lower bound or give FPT algorithms with dependency on k of the form $2^{o(k^2)}$.

Finally, while 'our' reductions are simple, the NP-hardness of Treewidth is derived from the NP-hardness of Pathwidth or Cutwidth. Thus, it would be good to have simple NP-hardness proofs for Pathwidth and/or Cutwidth, preferably building upon 'classic' NP-hard problems like Satisfiability, elementary graph problems like Clique, or Bin Packing.

Acknowledgments

We thank the reviewer for many useful comments.

This research was conducted in the Lorentz Center, Leiden, the Netherlands, during the workshop *Graph Decompositions: Small Width, Big Challenges*, October 24–28, 2022.

Dušan Knop and Ondřej Suchý acknowledge the support of the Czech Science Foundation Grant No. 22-19557S and by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590). Martin Milanič acknowledges partial support of the Slovenian Research and Innovation Agency (I0-0035, research program P1-0285 and research projects J1-3001, J1-3002, J1-3003, J1-4008, J1-4084, J1-60012, and N1-0370) and the research program CogniCom (0013103) at the University of Primorska. Sebastian Ordyniak acknowledges support by the Engineering and Physical Sciences Research Council (EPSRC, project EP/V00252X/1).

An extended abstract of this paper was published in the proceedings of the 18th International Symposium on Parameterized and Exact Computation (IPEC 2023) [5].

References

[1] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic and Discrete Methods*, 8(2):277–284, 1987. doi:10.1137/0608024.

[2] Stefan Arnborg and Andrzej Proskurowski. Characterization and recognition of partial 3-trees. *SIAM Journal on Algebraic Discrete Methods*, 7(2):305–314, 1986. doi:10.1137/0607033.

[3] Hans L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25:1305–1317, 1996. doi:10.1137/S0097539793251219.

[4] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.

[5] Hans L. Bodlaender, Édouard Bonnet, Lars Jaffke, Dusan Knop, Paloma T. Lima, Martin Milanic, Sebastian Ordyniak, Sukanya Pandey, and Ondrej Suchý. Treewidth

is NP-complete on cubic graphs. In Neeldhara Misra and Magnus Wahlström, editors, *Proceedings of the 18th International Symposium on Parameterized and Exact Computation, IPEC 2023*, volume 285 of *LIPIcs*, pages 7:1–7:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.IPEC.2023.7.

[6] Hans L. Bodlaender and Dimitrios M. Thilikos. Treewidth for graphs with small chordality. *Discrete Applied Mathematics*, 79(1-3):45–61, 1997. doi:10.1016/S0166-218X(97)00031-0.

[7] Édouard Bonnet. Treewidth inapproximability and tight ETH lower bound. In M. Koucký and N. Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025*, pages 2130–2135. ACM, 2025. doi:10.1145/3717823.3718117.

[8] John A. Ellis, Ivan Hal Sudborough, and Jonathan S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1):50–79, 1994. doi:10.1006/inco.1994.1064.

[9] Daniel J. Harvey and David R. Wood. The treewidth of line graphs. *Journal of Combinatorial Theory, Series B*, 132:157–179, 2018. doi:10.1016/j.jctb.2018.03.007.

[10] T. Kashiwabara and T. Fujisawa. NP-completeness of the problem of finding a minimum-clique-number interval graph containing a given graph as a subgraph. In *Proceedings International Symposium on Circuits and Systems*, pages 657–660, 1979.

[11] Nancy G. Kinnersley. The vertex separation number of a graph equals its path-width. *Information Processing Letters*, 42(6):345–350, 1992. doi:10.1016/0020-0190(92)90234-M.

[12] Tuukka Korhonen and Daniel Lokshtanov. An improved parameterized algorithm for treewidth. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, pages 528–541. ACM, 2023. doi:10.1145/3564246.3585245.

[13] Thomas Lengauer. Black-white pebbles and graph separation. *Acta Informatica*, 16:465–475, 1981. doi:10.1007/BF00264496.

[14] Fillia Makedon and Ivan Hal Sudborough. On minimizing width in linear layouts. *Discrete Applied Mathematics*, 23(3):243–265, 1989. doi:10.1016/0166-218X(89)90016-4.

[15] B. Monien and I. H. Sudborough. Min cut is NP-complete for edge weighted trees. *Theoretical Computer Science*, 58(1-3):209–229, 1988. doi:10.1016/0304-3975(88)90028-X.

[16] Tatsuo Ohtsuki, Hajimu Mori, Ernest S. Kuh, Toshinobu Kashiwabara, and Toshio Fujisawa. One-dimensional logic gate assignment and interval graphs. In *Proceedings of the IEEE Computer Society's Third International Computer Software and Applications Conference, COMPSAC 1979*, pages 101–106. IEEE, 1979. doi:10.1109/CMPSAC.1979.762474.

[17] Neil Robertson and Paul D. Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 45:92–114, 1986. doi:10.1016/0095-8956(86)90030-4.

[18] Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994. doi:10.1007/BF01215352.

[19] Yu (Ledell) Wu, Per Austrin, Toniann Pitassi, and David Liu. Inapproximability of treewidth and related problems. *Journal of Artificial Intelligence Research*, 49:569–600, 2014. doi:10.1613/JAIR.4030.