Twin-Width Meets Feedback Edges and Vertex Integrity

Jakub Balabán

Faculty of Informatics, Masaryk University, Brno, Czech Republic

Robert Ganian

Algorithms and Complexity Group, TU Wien, Vienna, Austria

Mathis Rocton

Algorithms and Complexity Group, TU Wien, Vienna, Austria

Abstract

The approximate computation of twin-width has attracted significant attention already since the moment the parameter was introduced. A recently proposed approach (STACS 2024) towards obtaining a better understanding of this question is to consider the approximability of twin-width via fixed-parameter algorithms whose running time depends not on twin-width itself, but rather on parameters which impose stronger restrictions on the input graph. The first step that article made in this direction is to establish the fixed-parameter approximability of twin-width (with an additive error of 1) when the runtime parameter is the feedback edge number.

Here, we make several new steps in this research direction and obtain:

- An asymptotically tight bound between twin-width and the feedback edge number;
- A significantly improved fixed-parameter approximation algorithm for twin-width under the same runtime parameter (i.e., the feedback edge number) which circumvents many of the technicalities of the original result and simultaneously avoids its formerly non-elementary runtime dependency;
- An entirely new fixed-parameter approximation algorithm for twin-width when the runtime parameter is the vertex integrity of the graph.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases twin-width, fixed-parameter algorithms, feedback edge number, vertex integrity

Digital Object Identifier 10.4230/LIPIcs.IPEC.2024.3

Related Version *Previous Version*: `https://arxiv.org/abs/2407.15514`

Funding *Robert Ganian*: Robert Ganian acknowledges support by the FWF and WWTF Science Funds (FWF project 10.55776/Y1329 and WWTF project ICT22-029).
Mathis Rocton: Mathis Rocton acknowledges support by the European Union's Horizon 2020 research and innovation COFUND programme (LogiCS@TUWien, grant agreement No 101034440), and the FWF Science Fund (FWF project Y1329).

1 Introduction

Twin-width is a comparatively recent graph-theoretic measure which is the culmination of as well as a catalyst for several recent breakthroughs in the area of algorithmic model theory [10, 11, 12, 13, 14]. Indeed, it has the potential to provide a unified explanation of why model-checking first order logic is fixed-parameter tractable on a number of graph classes which were, up to then, considered to be separate islands of tractability for the model-checking problem. This includes graphs of bounded rank-width, proper minor-closed graphs, map graphs [15], bounded-width posets [3] as well as a number of other specialized graph classes [4, 22].

19th International Symposium on Parameterized and Exact Computation (IPEC 2024).
Editors: Édouard Bonnet and Paweł Rzążewski; Article No. 3; pp. 3:1–3:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

While twin-width is related to graph parameters such as rank-width and path-width [14] as well as to measures which occur in matrix theory such as excluding linear minors [13], what distinguishes twin-width from these other measures is that we lack efficient algorithms for computing the twin-width of a graph. In particular, it is known that already deciding whether a graph has twin-width at most 4 is NP-hard [6]. This is highly problematic for the following reason: virtually every known algorithm that uses twin-width requires access to a so-called *contraction sequence*, which serves the same role as the decompositions typically used for classical parameters such as treewidth [36] and rank-width [35]. Intuitively speaking, a contraction sequence of width t – which serves as a witness for G having twin-width at most t – of a graph G is a sequence C of contractions of (not necessarily pairwise adjacent) vertex pairs which satisfies the following property: at each step of C, every vertex v only has at most t neighbors with an ancestor that is not adjacent to some ancestor of v[1].

The aforementioned NP-hardness of identifying graphs of twin-width 4 [6] effectively rules out fixed-parameter as well as XP algorithms for computing optimal contraction sequences when parameterized by the twin-width itself. One possible approach to circumvent this obstacle would be to devise a fixed-parameter algorithm which still uses the twin-width t as the parameter and computes at least an approximately-optimal contraction sequences, i.e., a contraction sequence of width $f(t)$ for some computable function f. On a complexity-theoretic level, such a result may be seen as "almost" as good as computing twin-width exactly, as it would still yield a fixed-parameter algorithm for first-order model checking.

Unfortunately, the task of finding such an algorithm has proven to be highly elusive, and it is far from clear that one even exists – in fact, whether twin-width can be approximated in fixed-parameter time (for any function f of the twin-width) can be seen as arguably the most prominent open question in contemporary research of twin-width. Recently [2], we attacked this question by first relaxing the running time requirement and ask whether we can obtain an $f(t)$-approximation for twin-width at least via a fixed-parameter algorithm where the runtime parameter is different (and, in particular, larger) than the twin-width t itself. As a first step in this direction, we developed a non-trivial fixed-parameter algorithm that computes a contraction sequence of width at most $t+1$ and is parameterized by the *feedback edge number* of the input graph, i.e., the edge deletion distance to acyclicity [2]. In the same paper, we also showed that the twin-width of a graph with feedback edge number k is upper-bounded by $k+1$.

Contributions. In this article, we significantly expand on our previous results [2] and present the next steps in the overarching program of understanding the boundaries of tractability for computing approximately-optimal contraction sequences. We summarize the three main contributions of this article below.

In Section 3, we revisit the relationship between twin-width and the feedback edge number. Here, we improve our previous linear bound [2] to square-root, and also show that this new bound is asymptotically tight. More precisely, we show that every graph class with feedback edge number k has twin-width $\mathcal{O}(\sqrt{k})$ (Theorem 8), and also construct a graph class with feedback edge number k whose twin-width is lower-bounded by $\Theta(\sqrt{k})$ (Proposition 9).

In Section 4, we revisit the main result of the preceding paper [2]: a polynomial-time reduction procedure which transforms every input graph G with feedback edge number k and twin-width t into a (tri-)graph G' whose twin-width lies between t and $t+1$ and whose size is upper-bounded by a non-elementary function of k. While this suffices to obtain the

[1] Formal definitions are provided in Section 2.

desired fixed-parameter approximation algorithm (as one may brute-force over all contraction sequences of G'), the dependence on the parameter k is astronomical and the proof relies on a sequence of highly technical arguments about how a hypothetical contraction sequence may be retrofitted in order to avoid certain degenerate steps. As the second main contribution of this article, we provide a new proof for the fixed-parameter approximability of twin-width parameterized by the feedback edge number which not only avoids many of the technical difficulties faced in the previous approach, but crucially also improves the size bound for the reduced instance G' from a *non-elementary* to a *quadratic* function of k.

Finally, in Section 5 we push the frontiers of approximability for twin-width by obtaining an algorithm which computes a contraction sequence for G of width at most twice the graph's twin-width and runs in time $f(p) \cdot |G|$, where p is the *vertex integrity* of G. Vertex integrity is a parameter which intuitively measures how easily a graph may be separated into small parts, and is defined as the smallest integer p such that there exists a separator X with the following property: each connected component C of $G - X$ satisfies $|V(C) \cup X| \leq p$. Vertex integrity may be seen as the natural intermediate step between the *vertex cover number* (which is the size of the smallest vertex cover in G, and which is known to allow for a trivial fixed-parameter algorithm for computing twin-width) and decompositional parameters such as *treedepth* and *treewidth* (for which the existence of a fixed-parameter approximation algorithm for twin-width remains a prominent open question [2]). Our result relies on a data reduction procedure which incorporates entirely different arguments than those used for the feedback edge number, and the correctness proof essentially shows that every optimal contraction sequence can be transformed into a near-optimal one where all "similar parts" of G are treated in a "similar way".

Related Work. Beyond the setting of computing twin-width and the associated contraction sequences, there are numerous other works which have targeted fixed-parameter algorithms for computing a structural graph parameter X when parameterized by graph parameters that differ from X. The general aim in this research direction is typically to further one's understanding of the fundamental problem of computing the targeted parameter X. Examples of fixed-parameter algorithms obtained in this setting include those for treewidth parameterized by the feedback vertex number [9], treedepth parameterized by the vertex cover number [33], MIM-width parameterized by the feedback edge number and other parameters [21], and the directed feedback vertex number parameterized by the (undirected) feedback vertex number [7]. The feedback edge number and vertex integrity have also been used to obtain parameterized algorithms for a number of other challenging problems [37, 5, 27, 34, 25, 29, 23, 30], whereas the latter parameter has also been studied in the literature under different asymptotically-equivalent names such as the *fracture number* [20, 24] and *starwidth* [38]. We refer interested readers to the very recent manuscript of Hanaka, Lampis, Vasilakis and Yoshiwatari [31] for a more detailed overview of vertex integrity and its relationship to other fundamental graph measures.

2 Preliminaries

For integers i and j, we let $[i, j] := \{n \in \mathbb{N} \mid i \leq n \leq j\}$ and $[i] := [1, i]$. We assume familiarity with basic concepts in graph theory [17] and parameterized algorithmics [18, 16]. When H is an induced subgraph of G, we denote it by $H \subseteq G$. Given vertex sets X and U, we will use $G[X]$ to denote the graph induced on X and $G - U$ to denote the graph $G[V(G) \setminus U]$; similarly, for an edge set F, $G - F$ denotes G after removing the edges in F.

A *dangling path* in G is a path of vertices which all have degree 2 in G, and a *dangling tree* in G is an induced subtree in G which can be separated from the rest of G by removing a single edge. The *length* of a path is the number of edges it contains. The *distance* between two vertices u and v is the length of the shortest path between them.

An edge set F in an n-vertex graph G is called a *feedback edge set* if $G - F$ is acyclic, and the *feedback edge number* of G is the size of a minimum feedback edge set in G. We remark that a minimum feedback edge set can be computed in time $\mathcal{O}(n)$ as an immediate corollary of the classical (DFS- and BFS-based) algorithms for computing a spanning tree in an unweighted graph G.

A graph has *vertex integrity* p if p is the smallest integer with the following property: G contains a vertex set S such that $S \neq V(G)$ and for each connected component H of $G - S$, $|V(H) \cup S| \leq p$. One may observe that the vertex integrity is upper-bounded by the size of a minimum vertex cover in the graph (i.e., the vertex cover number) plus one, and both vertex integrity and the feedback edge number are lower-bounded by treewidth minus one [36]. The vertex integrity of an n-vertex graph can be computed in time $\mathcal{O}(p^{p+1} \cdot n)$ [19].

Twin-Width. A *trigraph* G is a graph whose edge set is partitioned into a set of *black* and *red* edges. The set of red edges is denoted $R(G)$, and the set of black edges $E(G)$. The *black (*resp. *red) degree* of $u \in V(G)$ is the number of black (resp. red) edges incident to u in G. We extend graph-theoretic terminology to trigraphs by ignoring the colors of edges; for example, the degree of u in G is the sum of its black and red degrees (in the literature, this is sometimes called the *total degree*). We say a (sub)graph is *black (*resp. *red)* if all of its edges are black (resp. red); for example, P is a red path in G if it is a path containing only red edges. Without a color adjective, the path (or a different kind of subgraph) may contain edges of both colors.

Given a trigraph G, a *contraction* of two distinct vertices $u, v \in V(G)$ is the operation which produces a new trigraph by (1) removing u, v and adding a new vertex w, (2) adding a black edge wx for each $x \in V(G)$ such that $xu, xv \in E(G)$, and (3) adding a red edge wy for each $y \in V(G)$ such that $yu \in R(G)$, or $yv \in R(G)$, or y contains only a single black edge to either v or u. A sequence $C = (G = G_1, \ldots, G_n)$ is a *partial contraction sequence of* G if it is a sequence of trigraphs such that for all $i \in [n-1]$, G_{i+1} is obtained from G_i by contracting two vertices. A *contraction sequence* is a partial contraction sequence which ends with a single-vertex graph. The *width* of a (partial) contraction sequence C, denoted $w(C)$, is the maximum red degree over all vertices in all trigraphs in C. The *twin-width* of G, denoted $\text{tww}(G)$, is the minimum width of any contraction sequence of G, and a contraction sequence of width $\text{tww}(G)$ is called *optimal*. An example of a contraction sequence is provided in Figure 1.

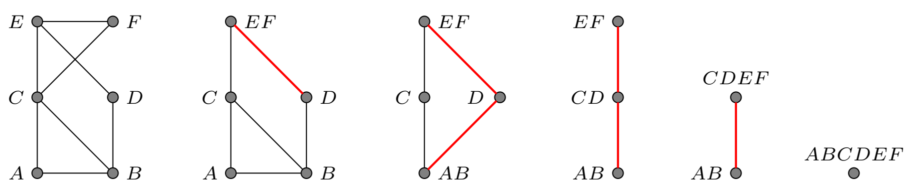

Figure 1 A contraction sequence of width 2 for the leftmost graph, consisting of 6 trigraphs.

Let us now fix a contraction sequence $C = (G = G_1, \ldots, G_n)$. For each $i \in [n]$, we associate each vertex $u \in V(G_i)$ with a set $\beta(u, i) \subseteq V(G)$, called the *bag* of u, which contains all vertices contracted into u. Formally, we define the bags as follows:

- for each $u \in V(G)$, $\beta(u, 1) := \{u\}$;
- for $i \in [n-1]$, if w is the new vertex in G_{i+1} obtained by contracting u and v, then $\beta(w, i+1) := \beta(u, i) \cup \beta(v, i)$; otherwise, $\beta(w, i+1) := \beta(w, i)$.

Note that if a vertex u appears in multiple trigraphs in C, then its bag is the same in all of them, and so we may denote the bag of u simply by $\beta(u)$. Let us fix $i, j \in [n]$, $i \leq j$. If $u \in V(G_i)$, $v \in V(G_j)$, and $\beta(u) \subseteq \beta(v)$, then we say that u is *an ancestor* of v in G_i and v is *the descendant* of u in G_j (clearly, this descendant is unique). If H is an induced subtrigraph of G_i, then $u \in V(G_j)$ is a *descendant* of H if it is a descendant of at least one vertex of H. A contraction of $u, v \in V(G_j)$ into $uv \in V(G_{j+1})$ *involves* $w \in V(G_i)$ if w is an ancestor of uv.

The following definition provides terminology that allows us to partition a contraction sequence into "steps" based on contractions between a subset of vertices in the original graph.

▶ **Definition 1.** *Let C be a contraction sequence of a trigraph G, and let H be an induced subtrigraph of G with $|V(H)| = m$. For $i \in [m-1]$, let $C\langle i\rangle_H$ be the trigraph in C obtained by the i-th contraction between two descendants of H, and let $C\langle 0\rangle_H = G$. For $i \in [m-1]$, let u_i and w_i be the two vertices that are contracted into the new vertex of $C\langle i\rangle_H$.*

A contraction sequence $C[H] = (H = H_1, \ldots, H_m)$ is the restriction of C to H *if for each $i \in [m-1]$, H_{i+1} is obtained from H_i by contracting the two vertices $u, w \in V(H_i)$ such that $\beta(u) = \beta(u_i) \cap V(H)$ and $\beta(w) = \beta(w_i) \cap V(H)$.*

It will also be useful to have an operation that forms the "reverse" of a restriction; we define this below.

▶ **Definition 2.** *Let G and H be graphs such that $H \subseteq G$ and let C_0 be a partial contraction sequence of H. We say that a partial contraction sequence C of G is the* extension *of C_0 to G if $C[H] = C_0$ and no contraction in C involves a vertex of $G - H$. When G_i is the i-th trigraph in C_0, we denote by $G_i \uparrow G$ the i-th trigraph in C (this makes sense since the lengths of C_0 and C are the same).*

Finally, we introduce a notion that will be useful when dealing with reduction rules in the context of computing contraction sequences.

▶ **Definition 3.** *Let G, G' be trigraphs. We say that the twin-width of G' is* effectively *at most the twin-width of G, denoted* $\text{tww}(G') \leq_e \text{tww}(G)$, *if* (1) $\text{tww}(G') \leq \text{tww}(G)$ *and* (2) *given a contraction sequence C of G, a contraction sequence C' of G' of width at most $w(C)$ can be constructed in polynomial time. If* $\text{tww}(G') \leq_e \text{tww}(G)$ *and* $\text{tww}(G) \leq_e \text{tww}(G')$, *then we say that the two graphs have* effectively *the same twin-width,* $\text{tww}(G') =_e \text{tww}(G)$.

Preliminary Observations and Remarks. We begin by stating a simple brute-force algorithm for computing twin-width.

▶ **Observation 4.** *An optimal contraction sequence of an n-vertex graph can be computed in time $2^{\mathcal{O}(n \cdot \log n)}$.*

Proof. Each contraction sequence is defined by $n-1$ choices of a pair of vertices, and so the number of contraction sequences is $\mathcal{O}((n^2)^n) = \mathcal{O}(2^{2n \cdot \log n}) \leq 2^{\mathcal{O}(n \cdot \log n)}$. Moreover, computing the width of a contraction sequence can clearly be done in polynomial time. ◀

The following observation provides a useful insight into the optimal contraction sequences of trees.

▶ **Observation 5** ([15, Section 3]). *For any rooted tree T with root r, there is a contraction sequence C of T of width at most 2 such that the only contraction involving r is the very last contraction in C.*

3 The Square-Root Bound

In this section, we prove that a graph with feedback edge number k has twin-width at most $\mathcal{O}(\sqrt{k})$. On a high level, the idea we will employ here builds on the preprocessing techniques originally introduced in the context of computing twin-width on tree-like graphs [2]: first we will contract the dangling trees, then the dangling paths, and for the final step we will use the following theorem of Ahn, Hendrey, Kim and Oum:

▶ **Theorem 6** ([1]). *If G is a graph with m edges, than the twin-width of G is at most $\sqrt{3m} + o(\sqrt{m})$.*

An issue we need to resolve before applying the aforementioned high-level approach is that Theorem 6 only applies to graphs without red edges, whereas the trigraph G we will obtain after dealing with the dangling trees and paths may contain these. The following lemma shows, using the properties of the red edges in G, that making all edges of G black can only decrease the twin-width by a constant.

▶ **Lemma 7.** *Let G be a trigraph with maximum red degree 2 such that each red edge in G is incident to a vertex of degree at most 2. If G' is the graph obtained from G by making all edges black, then* $\mathrm{tww}(G) \leq \mathrm{tww}(G') + 4$.

Proof. Let C' be an optimal contraction sequence of G', and let C be the contraction sequence of G obtained by following C'. We will prove that $w(C) \leq w(C') + 4$.

Let G_i be any trigraph in C and let G'_i be the trigraph in C' such that $V(G_i) = V(G'_i)$. Suppose for a contradiction that there are distinct vertices $u, v_1, v_2, v_3, v_4, v_5 \in V(G_i)$ such that for each $j \in [5]$, uv_j is a red edge in G_i but not in G'_i. Recall that $\beta(w)$ denotes the set of vertices contracted to w (the bag of w), and observe that for each $j \in [5]$, there must be vertices $u_j, v_j^0 \in V(G)$ such that $u_j \in \beta(u)$ and $v_j^0 \in \beta(v_j)$, and $u_j v_j^0$ is an edge that is black in G' but red in G. Since $uv_j \notin R(G'_i)$, there must be either all edges or no edges between $\beta(u)$ and $\beta(v_j)$ in G'. However, $u_j v_j^0 \in E(G')$, which means that for all $j, \ell \in [5]$, $u_j v_\ell^0$ is a black edge in G' (and so it is an edge also in G).

Since all vertices of G have red degree at most 2 and $u_j v_j^0 \in R(G)$ for each $j \in [5]$, there must be $a, b, c \in [5]$ such that $|\{u_a, u_b, u_c\}| = 3$. Now observe that each vertex in $\{u_a, u_b, u_c, v_a^0, v_b^0, v_c^0\}$ has degree at least 3 in G (since $u_j v_\ell^0$ is an edge in G for all $j, \ell \in \{a, b, c\}$). However, each red edge in G has an endpoint of degree at most 2, which is a contradiction.

We have proven that the red degree of each vertex $u \in V(G_i) = V(G'_i)$ may be higher in G_i than in G'_i by at most 4, which proves that $w(C) \leq w(C') + 4$. ◀

We are now ready to prove the square-root upper bound on twin-width.

▶ **Theorem 8.** *There exists a function $f(k) \in \mathcal{O}(\sqrt{k})$ such that every graph G with feedback edge number k has twin-width at most $f(k)$.*

Proof. Let F be a smallest feedback edge set of G and assume $k = |F| > 0$ (the case $k = 0$ follows from Observation 5). We will prove the statement by constructing a contraction sequence for G of width at most $f(k) \in \mathcal{O}(\sqrt{k})$. We begin by contracting each maximal dangling tree to a single vertex using Observation 5. After a maximal dangling tree has been

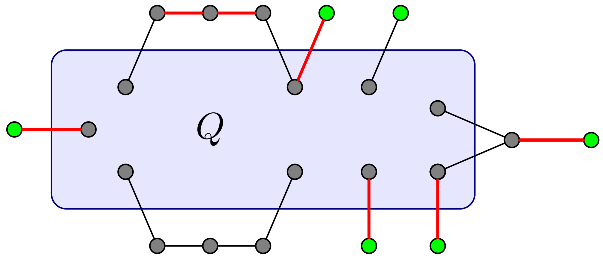

Figure 2 A trigraph after processing the dangling trees and shortening the paths in $\mathcal{P}$. Spikes are colored in green. The edges between vertices of Q are not depicted. Notice that one of the paths is black: this means it had no spikes and it has not been shortened. Also notice that one spike is attached by a black edge: it was a maximal dangling tree with only one vertex in G.

contracted, we call the last remaining vertex a *spike*, and we say that a vertex adjacent to a spike *has a spike*. Whenever a vertex has two spikes, we contract the spikes together (the obtained vertex is still called a spike). Observe that throughout this process, no vertex has red degree higher than 2: this is ensured by Observation 5 and the fact that a red neighbor of a vertex not in a dangling tree must be a spike.

Let G^α be the obtained trigraph and let T be the tree obtained from G^α by removing all spikes and edges in F. Let us choose any vertex of T to be the root. Let $Q_0 := \{u \in V(T) \mid u$ is incident to an edge of F in $G^\alpha\}$, $Q_1 := \{u \in V(T) \mid u$ has degree higher than 2 in $T\}$, and $Q := Q_0 \cup Q_1$. It is easy to see that $|Q_0| \leq 2k$ and that all leaves of T belong to Q_0 (a leaf not in Q_0 would belong to a dangling tree in G^α). Since a tree with n leaves has at most n vertices of degree higher than 2, we obtain that $|Q_1| \leq 2k$ and $|Q| \leq 4k$. Observe that $T - Q$ is a graph consisting of disjoint dangling paths. Let $\mathcal{P}$ be the set of these paths, and let $g\colon \mathcal{P} \to Q$ be the function such that $g(P)$ is the vertex of Q adjacent to the endpoint of P that is farther from the root of T. Since g is injective, we obtain that $|\mathcal{P}| \leq 4k$.

For each path $P = (u_1, \ldots, u_n)$ in $\mathcal{P}$, we perform the following contractions (starting with G^α).

- If $n > 2$, then for each $i \in [2, n-1]$ such that u_i has a spike v, contract u_i and v (do this in increasing order). If u_1 (resp. u_n) has a spike v, contract v and u_2 (resp. u_{n-1}).
- If $n > 3$, shorten P to a path with exactly three vertices by repeatedly contracting neighboring vertices of $P - \{u_1, u_n\}$.

Observe that throughout this process, no vertex has red degree higher than 2: a vertex in Q has red degree at most 1 (its red neighbor must be a spike) and a vertex in a path P either has a spike and at most one red neighbor in P or at most two red neighbors in P. See Figure 2 for an illustration.

Now we will count the number of edges in the obtained trigraph G^β. First, observe that there are at most $5k$ edges in $G^\beta[Q]$: k edges belonging to F and at most $4k$ other edges since $G^\beta[Q] - F$ is a forest. In addition, each vertex of Q may have a spike in G^β, which constitutes up to $4k$ other edges. Second, let $P \in \mathcal{P}$. If the length of P in G is at least 2, then P corresponds to at most four edges in G^β: at most two edges of the path itself and two edges connecting P to the rest of the graph (i.e., to vertices of Q). However, if P is shorter in G, then its vertices may have spikes in G^β and it may correspond to up to 5 edges: one edge of the path, two edges connecting it to Q, and two edges going to the spikes. Hence, $\mathcal{P}$ adds at most $20k$ edges, and thus there are at most $29k$ edges in G^β.

Let G^γ be the graph obtained from G^β by changing the color of all edges to black. By Theorem 6, G^γ has twin-width at most $\sqrt{87k} + o(\sqrt{k})$. Notice that G^β satisfies the preconditions of Lemma 7, which means that $\text{tww}(G^\beta) \leq \text{tww}(G^\gamma)+4$. Hence, the twin-width of G^β is also at most $\sqrt{87k} + o(\sqrt{k})$, and the same also holds for the original graph G (since the partial contraction sequence from G to G^β has width at most 2). ◀

We conclude the section by showing that Theorem 8 is asymptotically tight.

▶ **Proposition 9.** *There exists a function $f(k) \in \Omega(\sqrt{k})$ and an infinite class $\mathcal{G}$ of graphs such that for each $G \in \mathcal{G}$ with feedback edge number k, $\text{tww}(G) \geq f(k)$.*

Proof. Let n be a prime power such that $n \equiv 1 \pmod 4$. It is known that there exists an n-vertex $((n-1)/2)$-regular graph G (a so-called Paley graph) that has twin-width exactly $(n-1)/2$ [1, Section 3]. Since $|E(G)| = (n^2-n)/4$ and the spanning forest of G has at most $n-1$ edges, we know that the feedback edge number k of G is at least $(n^2-5n+4)/4 \in \Omega(n^2)$.

Let $\mathcal{G}$ be the class of all such n-vertex Paley graphs. For each n-vertex graph G in $\mathcal{G}$, we have $k \in \Omega(n^2)$ and $\text{tww}(G) \in \Theta(n)$. Let f be the function which maps each k to the minimum of $\{\text{tww}(G) \mid G \in \mathcal{G} \text{ is a graph with feedback edge number } k\}$. Thus, for each $G \in \mathcal{G}$, $\text{tww}(G) \geq f(k)$, and the aforementioned relationships between the number n of vertices of that graph, $\text{tww}(G)$ and k guarantee that $f(k) \in \Omega(\sqrt{k})$, as desired. ◀

4 A Better Algorithm Parameterized by the Feedback Edge Number

We begin by recalling that the case of twin-width 2 is known to already admit an exact nearly single-exponential fixed-parameter algorithm parameterized by the feedback edge number (see Theorem 10 below), and thus here we focus our efforts on graphs with higher twin-width.

▶ **Theorem 10** ([2])**.** *If G is a graph with feedback edge number k and $\text{tww}(G) \leq 2$, then an optimal contraction sequence of G can be computed in time $2^{\mathcal{O}(k \cdot \log k)} + n^{\mathcal{O}(1)}$.*

Our algorithm uses the same initial preprocessing steps as our previous result [2]. These are formalized through the following definition and theorem; note that in the approach we use here, we can use a slightly more general (and less technical) definition of *tidy $(H, \mathcal{P})$-graphs* than the preceding paper.

▶ **Definition 11.** *A connected trigraph G with $\text{tww}(G) \geq 2$ is a tidy $(H, \mathcal{P})$-graph if $\mathcal{P}$ is a non-empty set of dangling red paths in G, and there are two disjoint induced subtrigraphs of G, namely H and $\sqcup\mathcal{P}$ (the disjoint union of all paths in $\mathcal{P}$), such that each vertex of G belongs to one of them. Moreover, if $u \in V(H)$ has a neighbor $v \in V(\sqcup\mathcal{P})$ in G, then u has black degree 0 in G, and v is the only neighbor of u in $\sqcup\mathcal{P}$.*

The following theorem summarizes the results obtained in [2] that we will use in this section.

▶ **Theorem 12** ([2], Theorem 17 + Corollary 20)**.** *There is a polynomial-time procedure which takes as input a graph G with feedback edge number k and either outputs an optimal contraction sequence of G of width at most 2, or a tidy $(H, \mathcal{P})$-graph G' with effectively the same twin-width as G such that $|V(H)| \leq 112k$ and $|\mathcal{P}| \leq 4k$.*

From here on, we pursue an entirely different approach than the one used to obtain the previous (non-elementary) kernel [2]. In Subsection 4.1, we show how a tidy $(H, \mathcal{P})$-graph can be contracted when the paths in $\mathcal{P}$ are long enough and a contraction sequence of H is given. This is then used in Subsection 4.2, where we describe a better algorithm for approximating twin-width parameterized by the feedback edge number (see Theorem 18).

4.1 Contracting an $(H, \mathcal{P})$-Graph Using a Contraction Sequence for H

For this subsection, let us fix a tidy $(H, \mathcal{P})$-graph G and let $m := |\mathcal{P}|$. Assume that each $P \in \mathcal{P}$ satisfies $|V(P)| \geq 8m$ and let F be the subtrigraph of $\sqcup\mathcal{P}$ induced by the vertices at distance at most $2m$ from H in G.

Informally speaking, our goal now is to construct a "good" contraction sequence for such a trigraph G, see Corollary 16. To achieve that, we need to describe some well-structured trigraphs obtained by a sequence of contractions from G, which we will call *G-tidy trigraphs*, see the following Definition 13. An important property of a G-tidy trigraph is that all contractions happened either between two vertices of H or two vertices of F at the same distance from H (see items 1 and 3).

▶ **Definition 13.** *Let G' be a trigraph obtained by a sequence of contractions from G and let H' (resp. F') be the subtrigraph of G' induced by the vertices u such that $\beta(u)$ is a subset of $V(H)$ (resp. $V(F)$). We say that G' is a G-*tidy trigraph *if:*

1. *For $u \in V(G')$, we have $u \in V(H') \cup V(F')$ or $|\beta(u)| = 1$.*
2. *Each $u \in V(H')$ has at most one neighbor outside of H' in G'.*
3. *For each $u \in V(F')$, all vertices in $\beta(u)$ have the same distance d from H in G. We say that d is the* level *of u.*
4. *F' is a forest such that all its vertices have degree at most 3 in G'. If T is a connected component of F', then:*
   a. *T has exactly one vertex r at level 1 (let us declare it the root of T).*
   b. *The vertices of T with degree 3 in G_i form a subtree T' of T.*
   c. *The vertices of T' have level at most $|\beta(r)| - 1$, and either $T' = \emptyset$ or $r \in V(T')$.*

See Figure 3 for an illustration.

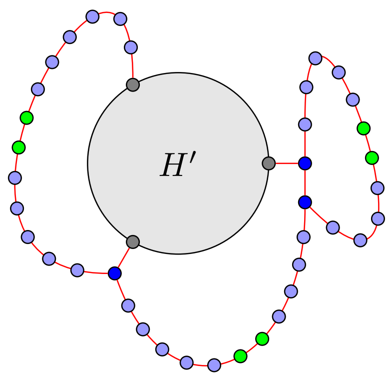

Figure 3 An illustration of Definition 13 when $m = 3$. The depicted G-tidy trigraph G' consists of H': vertices colored in grey, F': vertices colored in blue (degree-3 vertices in darker shade), and the remaining vertices are colored in green. The edges inside of H' are not depicted (there can be both red and black edges). Note that instead of each pair of green vertices, there should be at least 12 of them (because each path in $\mathcal{P}$ should contain at least $8m = 24$ vertices).

Now we show how G can be reduced to a G-tidy trigraph with H' being a single-vertex graph; this will be the first part of the proof of Corollary 16. Note that the assumption that C_H is given will be later handled in the proof of Lemma 17.

▶ **Lemma 14.** *Given a contraction sequence C_H of H, one can compute a partial contraction sequence of width $\max(w(C_H) + 1, 4)$ from G to a G-tidy trigraph G' with $|V(H')| = 1$, in polynomial time.*

Proof. For each $i \in [|V(H)|]$, we will construct a partial contraction sequence C_i from G to a G-tidy trigraph G_i such that $w(C_i) \leq \max(w(C_H)+1, 4)$ and the restriction of C_i to H will be the prefix of C_H of length i. We will denote the subtrigraphs of G_i corresponding to H' and F' (see Definition 13) by H_i and F_i, respectively. We define $C_1 = (G)$, i.e., C_1 is the trivial partial contraction sequence with no contractions. It can be easily verified that $G_1 := G$ is a G-tidy trigraph. In particular, the forest $F_1 := F$ consists of $2m$ disjoint paths.

Suppose that we have constructed C_i for some $i < |V(H)|$. Let $u, v \in V(H_i)$ be the two vertices contracted in H_{i+1} (which is the successor of H_i in C_H). If u or v does not have a neighbor outside of H_i in G_i, then we define G_{i+1} to be the trigraph obtained from G_i by contracting u and v. Clearly, G_{i+1} is a G-tidy trigraph and C_{i+1} (the sequence obtained by prolonging C_i with G_{i+1}) has the required properties. Now suppose that both u and v have a neighbor outside of H_i in G_i. In this case, we cannot simply contract them because the new vertex would have two neighbors outside of H_{i+1}, violating condition 2 of Definition 13.

Let T_u and T_v be the two connected components of F_i with roots adjacent to u and v, respectively. Informally, we need to merge T_u and T_v before we can contract u and v. Let $T \in \{T_u, T_v\}$ be a tree with root r. If T contains no degree-3 vertices, we do nothing (we always mean degree in G_i). Otherwise, let $w \in V(T)$ be the deepest degree-3 vertex such that all its ancestors in T have degree 3. By item 3 of Definition 13, $|\beta(r)| \leq 2m$ because F contains exactly $2m$ vertices at level 1 (2 for each path in $\mathcal{P}$). Hence, by item 4c, the level of w is less than $2m$, and so w has two children x and y in T, both of degree 2. We contract x and y (note that the obtained vertex xy has red degree 3, and the red degree of w drops to 2). We repeat this process as long as such vertex w exists (crucially, xy cannot be chosen as the next w because its parent has degree 2). Afterwards, we contract the roots r_u, r_v of T_u and T_v, and finally, we contract u and v into uv.

Let C_{i+1} be the partial contraction sequence of G obtained by prolonging C_i with the contractions described in the previous paragraph. Let us show that $w(C_{i+1}) \leq \max(w(C_H)+1, 4)$. By the assumption about C_i, it suffices to discuss red degrees in each trigraph G' between G_i and G_{i+1} (which is the last trigraph in C_{i+1}). Clearly, any descendant of H in G' has red degree at most $w(C_H)+1$ (it is crucial that u and v are contracted after r_u and r_v). Any other vertex of G' has red degree at most 3, except for the vertex obtained by contracting r_u and r_v, whose red degree is 4 (but it drops to 3 when u and v are contracted).

Finally, we need to show that G_{i+1} is G-tidy. It is easy to see that G_{i+1} satisfies the first three items of Definition 13. To prove that G_{i+1} satisfies item 4, observe that F_{i+1} is indeed a forest: it contains the same trees as F_i, except that T_u and T_v have been merged into a new tree T with root r. More precisely, T is isomorphic to the tree obtained from the disjoint union of T_u and T_v by first adding a new vertex r and edges rr_u, rr_v, and second removing all leaves. Since r has degree 3 in G_{i+1}, the highest level of a degree-3 vertex in T is one higher than in the union of T_u and T_v in G_i. Since $|\beta(r)| = |\beta(r_u)| + |\beta(r_v)|$ and both of these summands are at least 1, we get that G_{i+1} satisfies item 4c, which concludes the proof. ◀

To prove Corollary 16, we now show that the G-tidy trigraph given by Lemma 14 can be contracted to a single vertex (without creating vertices with high red degree). Note that the following proof is inspired by the proof of Theorem 7 in [6].

▶ **Lemma 15.** *If G' is a G-tidy trigraph G' with $|V(H')| = 1$, then a contraction sequence of G' of width at most 4 can be computed in polynomial time.*

Proof. First, observe that by Definition 11, all edges in G' are red. By item 2 of Definition 13, the only vertex u of H' has a single neighbor r. By Definition 11, G is connected; hence, also G' is connected. This implies that F' is a tree. Let T be the subtree of F' induced by the vertices with degree 3 in G'. Let us begin by contracting u and r, obtaining a trigraph $G^* := G' - u$.

Observe that the depth of T is at most $2m - 1$ because r contains $2m$ vertices in its bag (by property 4c). Consider a path $P \in \mathcal{P}$ and observe that the descendants of at most $4m$ vertices of P belong to T in G^* ($2m$ from each side). Hence, $G^* - T$ consists of disjoint dangling red paths, each with at least $4m$ vertices (since each $P \in \mathcal{P}$ satisfies $|V(P)| \geq 8m$). Let $\mathcal{P}'$ be the set of these red paths in G^*.

Let $P \in \mathcal{P}'$ and let u and u' be the endpoints of P. Let $v, v' \in V(T)$ be the neighbors of u and u' in T, respectively, and let Q be the path connecting v and v' in T. Since the depth of T is at most $2m - 1$, we know that Q contains at most $4m - 1$ vertices. Let us shorten P so that it has the same length as Q (by repeatedly contracting consecutive vertices). Let $(u_1 = u, \dots, u_p = u')$ and $(v_1 = v, \dots, v_p = v')$ be the sequences of vertices of P and Q in the natural orders. Now for each $i \in [p]$ in increasing order, contract u_i and v_i, and observe that the obtained trigraph is isomorphic to $G^* - P$. Repeat this for all paths $P \in \mathcal{P}'$, obtaining a trigraph isomorphic to T, which has twin-width at most 3 and can be contracted as per Observation 5. Finally, observe that during a contraction of a path in $P \in \mathcal{P}'$, there is never a vertex with red degree higher than 4. Indeed, after contracting u_i and v_i for $i \in [p-1]$, the obtained vertex has at most four red neighbors: at most three in T plus u_{i+1}. ◀

4.2 Wrapping up the Proof

In the previous subsection, we proved Lemmas 14 and 15, which together imply the following corollary.

▶ **Corollary 16.** *Let G be a tidy $(H, \mathcal{P})$-graph such that each $P \in \mathcal{P}$ satisfies $|V(P)| \geq 8 \cdot |\mathcal{P}|$. Given a contraction sequence C_H of H, one can compute a contraction sequence of G of width $\max(w(C_H) + 1, 4)$, in polynomial time.*

Now we are able to show that if we shorten all long paths in a tidy $(H, \mathcal{P})$-graph, then the twin-width increases by at most 1 (formally, shortening a path means contracting its consecutive vertices).

▶ **Lemma 17.** *Let G be a tidy $(H_0, \mathcal{P}_0)$-graph such that $\operatorname{tww}(G) \geq 3$, let $m = |\mathcal{P}_0|$ and let G' be the trigraph obtained from G by shortening each path $P \in \mathcal{P}_0$ with more than $8 \cdot m$ vertices to length exactly $8 \cdot m - 1$. Then $\operatorname{tww}(G') \leq \operatorname{tww}(G) + 1$.*

Proof. We begin by handling short paths in $\mathcal{P}_0$: let $\mathcal{P}_{short} = \{P \in \mathcal{P}_0 \colon |V(P)| < 8m\}$, let H be the union of H_0 and $\sqcup \mathcal{P}_{short}$ (including the edges between them), and let $\mathcal{P} = \mathcal{P}_0 \setminus \mathcal{P}_{short}$. Clearly, G is also a tidy $(H, \mathcal{P})$-graph. Also observe that G' is a tidy $(H, \mathcal{P}')$-graph (where $\mathcal{P}'$ is the set of paths obtained from $\mathcal{P}$ by shortening each path in it).

We want to construct a contraction sequence C' of G' of width at most $\operatorname{tww}(G) + 1$ from an optimal contraction sequence C of G. Let C_H be the restriction of C to H; clearly, $w(C_H) \leq \operatorname{tww}(G)$. Since $\operatorname{tww}(G) \geq 3$, it suffices to apply Corollary 16 on G' using C_H, which yields the desired contraction sequence C'. ◀

Finally, we are able to prove the main result of this section.

▶ **Theorem 18.** *Given a graph G with feedback edge number k, a trigraph G' of size $\mathcal{O}(k^2)$ such that $\operatorname{tww}(G) \leq \operatorname{tww}(G') \leq \operatorname{tww}(G) + 1$ can be computed in polynomial time. Moreover, a contraction sequence for G of width at most $\operatorname{tww}(G) + 1$ can be computed in time $2^{\mathcal{O}(k^2 \cdot \log k)} + n^{\mathcal{O}(1)}$.*

Proof. First, we use Theorem 10 to check whether $\text{tww}(G) \leq 2$ (if yes, G' can be any constant-size graph with the same twin-width as G). From now on, assume $\text{tww}(G) \geq 3$. Now let us use Theorem 12 to obtain a tidy $(H, \mathcal{P})$-graph G_1 with effectively the same twin-width as G such that $|V(H)| \leq 112k$ and $|\mathcal{P}| \leq 4k$. Let G' be the trigraph obtained when Lemma 17 is applied on G_1. By Lemma 17, $\text{tww}(G') \leq \text{tww}(G_1) + 1$. Conversely, $\text{tww}(G') \geq \text{tww}(G_1)$ because there is a partial contraction sequence C_1 from G_1 to G' of width at most $\text{tww}(G')$; it suffices to shorten paths of $\mathcal{P}$ that are shorter in G' than in G_1 by contracting consecutive vertices. Hence, we indeed have $\text{tww}(G) \leq \text{tww}(G') \leq \text{tww}(G) + 1$.

Next, let us examine the size of G'. By Lemma 17, each of the $4k$ paths in $\mathcal{P}$ has at most $8 \cdot 4k$ vertices in G'. Hence, we obtain $|V(G')| \leq 128k^2 + 112k \in \mathcal{O}(k^2)$ as required.

Finally, let us show how a contraction sequence for G of width at most $\text{tww}(G) + 1$ can be computed. If $\text{tww}(G) \leq 2$, then this contraction sequence is provided by Theorem 10. Otherwise, observe that an optimal contraction sequence C' of G' can be computed in time $2^{\mathcal{O}(k^2 \cdot \log k)}$ by Observation 4. Next we concatenate C' and C_1 (which is defined above and can be computed trivially) to obtain a contraction sequence of G_1 of width at most $\text{tww}(G_1) + 1$. We conclude using the effectiveness part of $\text{tww}(G) =_e \text{tww}(G_1)$ (see Definition 3). ◀

5 A Fixed-Parameter Algorithm Parameterized by Vertex Integrity

In this section, we design an FPT 2-approximation algorithm for computing twin-width when parameterized by the vertex integrity, see Theorem 23.

5.1 Initial Setup and Overview

For the following, it will be useful to recall the definition of vertex integrity presented in Section 2. Let us fix a graph G and a choice of $S \subseteq V(G)$ witnessing that the vertex integrity of an input graph G is p, and let $\mathcal{C}$ be the set of connected components of $G - S$. We assume without loss of generality that G is connected, as the twin-width of a graph is the maximum twin-width of its connected components. We now define a notion of "component-types" which intuitively captures the equivalence between components which exhibit the same outside connections and internal structure.

▶ **Definition 19.** *We say that two graphs $H_0, H_1 \in \mathcal{C}$ are* twin-blocks*, denoted $H_0 \sim H_1$, if there exist a* canonical *isomorphism α from H_0 to H_1 such that for each vertex $u \in V(H_0)$ and each $v \in S$, $uv \in E(G)$ if and only if $\alpha(u)v \in E(G)$. Clearly, $\sim$ is an equivalence relation.*

In a nutshell, our algorithm first computes an optimal contraction sequence C' for a subgraph G' of G that is obtained by keeping only a bounded number of twin-blocks from each equivalence class, and then uses C' to obtain a contraction sequence for G of width at most $2 \cdot \text{tww}(G') \leq 2 \cdot \text{tww}(G)$. In the following definition, we introduce terminology related to subgraphs of G.

▶ **Definition 20.** *Let G' be an induced subgraph of G.*

- *We say that G' is $\mathcal{C}$-*respecting *if $S \subseteq V(G')$ and for each $H \in \mathcal{C}$, either $H \subseteq G'$ or $V(H) \cap V(G') = \emptyset$.*
- *We say that an equivalence class $[H_0]$ of $\sim$ is* large *in G' if $|\mathcal{H}| \geq f(p)$, where $\mathcal{H} = \{H \in [H_0] \mid H \subseteq G'\}$ and $f(p) = 2^{7p^3}$.*
- *We say that G' is the* reduced graph *of G if it is obtained from G by removing all but $f(p)$ twin-blocks from each large class of $\sim$.*

Let us now bound the size of the reduced graph G'.

▶ **Observation 21.** *If G' is the reduced graph of G, then $|V(G')| \leq p + p^2 \cdot f(p) \cdot 2^{2p^2} \in 2^{\mathcal{O}(p^3)}$.*

Proof. First, let us compute the size of $\mathcal{C}/{\sim}$. Each $H \in \mathcal{C}$ has at most p vertices, which means that the number of non-isomorphic graphs in $\mathcal{C}$ can be upper-bounded by $p \cdot 2^{p^2}$. Since $|S| \leq p$, there are at most p^2 possible edges between S and each $H \in \mathcal{C}$. Hence, $|\mathcal{C}/{\sim}| \leq p \cdot 2^{2p^2}$. Because $|V(H)| \leq p$ for each $H \in \mathcal{C}$ and by definition of G', the union of each class of $\sim$ contains at most $p \cdot f(p)$ vertices. Finally, we again use that $|S| \leq p$. ◀

The core of our algorithm is the following lemma, which we will prove in Subsection 5.2:

▶ **Lemma 22.** *If G' is the reduced graph of G, then given a contraction sequence C' for G' of width t, we can compute a contraction sequence for G of width at most $2t$ in polynomial time.*

Let us now show how we can use this lemma to design the desired algorithm:

▶ **Theorem 23.** *If G is a graph with vertex integrity p, then a contraction sequence for G of width at most $2 \cdot \operatorname{tww}(G)$ can be computed in time $g(p) \cdot n^{\mathcal{O}(1)}$, where g is an elementary function.*

Proof. The first step of the algorithm is to compute an optimal vertex-integrity decomposition of G. As noted already in Section 2, this can be done in time $\mathcal{O}(p^{p+1} \cdot n)$ [19]. Using this decomposition, we can compute the reduced graph G' of G in polynomial time. Next, we can compute an optimal contraction sequence C' of G', using Observation 4. Since the size of G' is bounded (see Observation 21), we deduce that computing C' takes time $g(p) \in \exp(\exp(\mathcal{O}(p^3)))$, where $\exp(x) = 2^x$.

Finally, we apply Lemma 22 to compute in polynomial time a contraction sequence C for G of width at most $2 \cdot w(C') = 2 \cdot \operatorname{tww}(G')$. Since G' is an induced subgraph of G, we know $\operatorname{tww}(G') \leq \operatorname{tww}(G)$, which implies the desired bound $w(C) \leq 2 \cdot \operatorname{tww}(G)$. ◀

5.2 Extending a contraction sequence from G' to G

This subsection is dedicated to proving Lemma 22. Recall that we have fixed a graph G and a set $S \subseteq V(G)$, and that $\mathcal{C}$ is the set of connected components of $G - S$. Let us begin with several technical definitions.

▶ **Definition 24.** *Let G' be a $\mathcal{C}$-respecting graph, let H_0 and H_1 be distinct twin-blocks (with canonical isomorphism α) such that $H_0, H_1 \subseteq G'$, and let G^* be any trigraph obtained from G' by a sequence of contractions. We say that H_0 and H_1 are* merged *in G^* if, for each $u \in V(H_0)$, there is a vertex $v \in V(G^*)$ such that $u, \alpha(u) \in \beta(v)$.*

It might be confusing that in the following definition, we consider a $\mathcal{C}$-respecting graph and a graph $H \in \mathcal{C}$ that is *not* its induced subgraph. The reason for this is that later we will show that, under some conditions, H can be "added" without increasing the twin-width too much. In fact, all such graphs H will be progressively added until all of them are present (and the obtained graph is the whole G). To formalize the process of adding H, we will use Definition 2 to create an extension of a contraction sequence to a sequence with H "appended" to all trigraphs.

▶ **Definition 25.** *Let G' be a $\mathcal{C}$-respecting graph, let $H \in \mathcal{C}$ be such that $H \not\subseteq G'$, let $C' = (G'_1, G'_2, \ldots)$ be a contraction sequence of G'.*

- *We say that a trigraph G'_i in C' is the C'-*critical trigraph *for H if i is the least index such that some vertex of H has a red neighbor in $G'_i \uparrow G$.*
- *If G'_i is the C'-critical trigraph for H, then we say that a trigraph G'_j is C'-*safe *for H if $j < i$ and there are two graphs $H', H'' \in [H]_\sim$ that are merged in G'_j.*

We will show that for each H and C' (as in Definition 25), there is a C'-safe trigraph for H. The first step towards this is to show that if H has many twin-blocks in G', then there are two twin-blocks of H merged in the C'-critical trigraph G^* for H. Intuitively, if the twin-blocks of H were not "sufficiently-merged" in G^*, then some vertex of S would have high red degree because the existence of a red edge between S and H (see the definition of C'-critical) implies red edges between S and all twin-blocks of H.

▶ **Lemma 26.** *If G' is a $\mathcal{C}$-respecting graph, C' is a contraction sequence of G', $H \in \mathcal{C}$ is a graph such that $H \not\subseteq G'$, the class $\mathcal{H} := [H]_\sim$ is large in G', and G^* is the C'-critical trigraph for H, then there are two graphs $H', H'' \in [H]_\sim$ that are merged in G'_i.*

Proof. Let $I = [f(p)]$ and let $H_1, \ldots, H_{f(p)} \in \mathcal{H}$ be distinct graphs such that $H_i \subseteq G'$ for each $i \in I$ (using the fact that $\mathcal{H}$ is large in G'). For $i \in I$ and $u \in V(H)$, let $u_i := \alpha(u)$, where $\alpha\colon V(H) \to V(H_i)$ is a canonical isomorphism. Let $u, v \in V(G^* \uparrow G)$ be two vertices such that $u \in V(H)$ and uv is a red edge in $G^* \uparrow G$. By Definition 25, such vertices u and v exist, and by definition of vertex integrity, v is a descendant of S. Let $d := 2^{p+1} + 1$. We shall prove by induction that the following claim holds.

▷ Claim 27. For each $a \in [0, p-1]$, there is a set $I_a \subseteq I$ of size at least $f(p)/d^{pa+1}$ such that for each $i, j \in I_a$ and each vertex $w \in V(H)$ at distance at most a from u in H, there is a vertex $x \in V(G^*)$ such that $w_i, w_j \in \beta(x)$.

Observe that this statement implies that H_i and H_j for any $i, j \in I_{p-1}$ are merged in G^* because the diameter of H is at most $p - 1$.

Proof of Claim 27. Let us start by proving Claim 27 for $a = 0$. Let $U = \{u_i \mid i \in I\}$ and observe that for each $i \in I$, the descendant u'_i of u_i is a red neighbor of v in G^*, by Definition 19 (unless $u'_i = v$). However, the red degree of v in G^* is at most $\mathrm{tww}(G') \le 2^{p+1}$ (because the treewidth of G' is at most the vertex integrity of G', and the twin-width is bounded by treewidth, see [32]). Hence, the vertices of U are present in the bags of at most d vertices in G^* (note that some vertices of U may be in the bag of v), which means that there is a vertex $w \in V(G^*)$ with at least $f(p)/d$ vertices of U in its bag. Now it suffices to set $I_0 := \{i \in I \mid u_i \in \beta(w)\}$. This concludes the proof of the base case of the induction.

For the induction step, suppose that Claim 27 holds for some $a \in [0, p-2]$, i.e., there is a set $I_a \subseteq I$ with the described properties. Let $D_a, D_{a+1} \subseteq V(H)$ be the sets of vertices at distance exactly a or $a+1$ from u in H, respectively. Let $w \in D_{a+1}$ and $x \in D_a$ be two neighbors in H. Let x' be the descendant of x_i in G^* for some $i \in I_a$ (or, equivalently, for each $i \in I_a$, by the induction hypothesis), let $W = \{w_i \mid i \in I_a\}$, and let w'_i be the descendant of w_i in G^* (for any $i \in I_a$). Observe that $x'w'_i$ is a red edge of G^+, unless $x' = w'_i$. Using the same argument as in the base case, x' has red degree at most $d - 1$ in G^*, which means that the vertices of W are present in the bags of at most d vertices in G^*.

Since $|D_{a+1}| \le p$, I_a can be partitioned into at most d^p parts such that if $i, j \in I_a$ are in the same part, then for each vertex $w \in D_{a+1}$, w_i and w_j are in the bag of the same vertex in G^*. Hence, one of these parts has size at least $|I_a|/d^p$, and we choose it to be I_{a+1}. A simple computation shows that I_{a+1} satisfies Claim 27. ◁

Finally, we only need to verify that $|I_{p-1}| \geq f(p)/d^{p(p-1)+1} \geq 2$. Recall that $f(p) = 2^{7p^3}$ and $d = 2^{p+1} + 1 \leq 2^{3p}$ since $p \geq 1$. Since $p(p-1) + 1 \leq 2p^2$, we get $|I_{p-1}| \geq 2^{7p^3}/2^{6p^3} \geq 2$, which concludes the proof. ◀

Now we need to take a closer look at S.

▶ **Definition 28.** *Let $H \in \mathcal{C}$ and $u, v \in S$. We say that u and v are* H-equivalent*, denoted $u \sim_H v$, if and only if for each $w \in V(H)$, $uw \in E(G) \Leftrightarrow vw \in E(G)$. Let $S^H \subseteq S$ be the set of vertices with at least one neighbor in H (in G). If G'_i is a trigraph in a contraction sequence of a $\mathcal{C}$-respecting graph, then we denote by S^H_i the set of descendants of S^H in G'_i.*

A crucial observation is that before the C'-critical trigraph for H, only very restricted contractions may involve vertices of S^H (so that a red edge to H does not appear).

▶ **Observation 29.** *If G' is a $\mathcal{C}$-respecting graph, $C' = (G'_1, G'_2, \dots)$ is a contraction sequence of G', $H \in \mathcal{C}$ is a graph such that $H \not\subseteq G'$, G'_i is the C'-critical trigraph for H, and $j < i$, then for each $u \in S^H_j$, the bag $\beta(u)$ is a subset of an equivalence class of $\sim_H$.*

Proof. Suppose for contradiction that there is $u \in S^H_j$ such that $\beta(u)$ is not a subset of an equivalence class of $\sim_H$. If $\beta(u) \not\subseteq S$, then clearly all neighbors of u in H (in $G'_j \uparrow G$) would be red, a contradiction with $j < i$ and the choice of i. Hence, assume $\beta(u) \subseteq S$. If there are $v_0, v_1 \in \beta(u)$ such that $v_0 \nsim_H v_1$, then there is a vertex $w \in H$ that has exactly one neighbor in $\{v_0, v_1\}$ in G, by Definition 28. Thus, uw is a red edge in $G'_j \uparrow G$, again a contradiction ◀

Using Observation 29, we can prove the existence of a C'-safe trigraph.

▶ **Lemma 30.** *If G', C', H and G'_i are as in Observation 29 and the equivalence class $[H]_\sim$ is large in G', then G'_{i-1} is a C'-safe trigraph for H.*

Proof. By Definition 25, it suffices to show that there are two graphs $H', H'' \in [H]_\sim$ that are merged in G'_{i-1}. By Lemma 26, we know that such merged graphs H' and H'' exist for G'_i. Let $u, v \in V(G'_{i-1})$ be the two vertices that are contracted in G'_i, and suppose for contradiction that H' and H'' are not merged already in G'_{i-1}. This implies that u and v are both descendants of $H' \cup H''$. However, by Observation 29, $u, v \notin S^H_{i-1}$. This is a contradiction with Definition 25 because the contraction creating G'_i must involve a vertex of S^H so that a red edge incident to H can appear in $G'_i \uparrow G$. ◀

Now we are ready to show how a contraction sequence C' of G' can be modified when a graph $H \in \mathcal{C}$ is added to G'. Unfortunately, we cannot do that without increasing the width. Since our goal is to eventually add many graphs $H \in \mathcal{C}$, we need to keep the increase under control, for which we use the following definition.

▶ **Definition 31.** *A contraction sequence $C = (G_1, \dots, G_n)$ has* progressive width $(a \to_i b)$ *if the width of $(G_1, \dots, G_{i-1})$ is at most a and the width of $(G_i, \dots, G_n)$ is at most b.*

▶ **Lemma 32.** *Let G' be a $\mathcal{C}$-respecting graph, let $C' = (G'_1, G'_2, \dots)$ be a contraction sequence of G', let $H \in \mathcal{C}$, $H \not\subseteq G'$ be such that $\mathcal{H} := [H]_\sim$ is large in G', let $G^+ = G[V(G') \cup V(H)]$, and let G'_{i-1} be a C'-safe trigraph for H. If C' has progressive width $(t \to_i 2t)$, then we can construct in polynomial time a contraction sequence C^+ for G^+ of progressive width $(t \to_i 2t)$. Moreover, if $j < i$, then $G'_j \uparrow G^+$ is the j-th trigraph in C^+.*

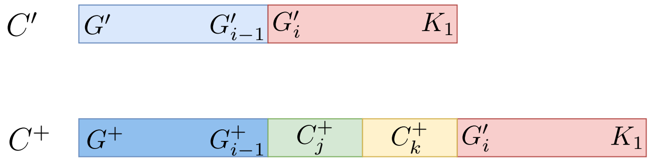

Figure 4 A schematic depiction of the construction of C^+ from C' in the proof of Lemma 32. Informally, we insert a new contraction segment after G'_{i-1} (the green and the yellow block), which handles H. The blue prefixes of the two contraction sequences are "morally" the same but in C^+, H is still present, and so G'_ℓ is not isomorphic to G^+_ℓ for $\ell \in [i-1]$ but it is an induced subtrigraph thereof. On the other hand, the red suffixes are exactly the same since H has been contracted with H'.

Proof. By Definition 25, there are $H', H'' \in \mathcal{H}$ (such that $H', H'' \subseteq G'$) that are merged in G'_{i-1}. Let $\iota\colon H \to H'$ be a canonical isomorphism, let $C'_{<i}$ be the prefix of C' of length $i-1$, and let C_H be the partial contraction sequence of H isomorphic to $C'_{<i}[H']$ with an isomorphism induced by ι[2]. Let us now construct $C^+ = (G^+_1 = G^+, G^+_2, \ldots)$; see also Figure 4 for an illustration:

1. $C^+_i := (G^+_1, \ldots, G^+_{i-1})$ is the extension of $C'_{<i}$ to G^+, i.e., the same contractions are performed, ignoring H. Note that this construction shows that $G'_j \uparrow G^+$ for each $j < i$, as required.
2. $C^+_j := (G^+_{i-1}, \ldots, G^+_j)$ is the extension of C_H to G^+_{i-1}, i.e., C_H is applied to H, ignoring the rest of G^+_{i-1}.
3. Let H_j and H'_j be the subtrigraphs of G^+_j induced by the descendants of H and H', respectively. By Definition of C_H, there is a bijection α_j from $V(H_j)$ to $V(H'_j)$ that respects[3] ι. Let $C^+_k := (G^+_j, \ldots, G^+_k)$ be the contraction sequence that contracts u and $\alpha_j(u)$ for every $u \in V(H_j)$ in arbitrary order.
4. We will prove that $G^+_k \cong G'_{i-1}$, and we will define the rest of C^+ to be the suffix of C' starting with G'_i.

Let us argue that C^+ can be computed in polynomial time. First, we need to find the two merged graphs $H', H'' \in \mathcal{H}$: this can be done by brute force because the size of $\mathcal{H}$ is at most $\mathcal{O}(n)$ and checking whether given H' and H'' are merged can be done efficiently (the details depend on the computational model and the representation of contraction sequences). Then, we compute C_H by going through $C'_{<i}$ and looking only at contractions involving vertices of H'. Using C_H, it is easy to compute C^+_j. All other parts of C^+ can clearly be computed in polynomial time.

Now we need to show that C^+ has progressive width $(t \to_i 2t)$. By the assumption about the progressive width of C', C^+_i has width at most t (using also the fact that G'_{i-1} is a C'-safe trigraph for H; no red edge in C^+_i is incident to H). Hence, we only need to prove that the suffix of C^+ starting with G^+_i has width at most $2t$. Let S^H_j be the set containing the descendants S^H in G^+_j (or, equivalently, in G'_{i-1}, G^+_{i-1} or G^+_k).

[2] Formally, an isomorphism from $(G = G_1, \ldots, G_n)$ to $(H = H_1, \ldots, H_n)$ induced by an isomorphism $\alpha\colon G \to H$ is a sequence of isomorphisms $\alpha_i\colon G_i \to H_i$ such that for each $i \in [n]$ and $u \in V(G_i)$, $\beta(u) = \alpha^{-1}(\beta(\alpha_i(u)))$.

[3] By respecting ι, we mean that if $u \in \beta(v)$ for $u \in V(H)$, $v \in V(H_j)$, then $\iota(u) \in \beta(\alpha_j(v))$.

▷ Claim 33. C_j^+ has width at most $2t$. Moreover, descendants of H have red degree at most t in trigraphs of C_j^+.

Proof of the Claim. Let $\ell \in [i, j]$, let H_ℓ be the subtrigraph of G_ℓ^+ induced by the descendants of H, and let $m \in [i-1]$ be an index such that the subtrigraph H'_m of G_m^+ induced by the descendants of H' satisfies $|V(H_\ell)| = |V(H'_m)|$. We need to show that the red degree of each $u \in V(G_\ell^+)$ is at most $2t$ (and at most t when $u \in V(H_\ell)$). By construction of C_j^+, there is a bijection $\alpha : V(H_\ell) \to V(H'_m)$ such that if $u \in \beta(v)$ for $u \in V(H)$, $v \in V(H_\ell)$, then $\iota(u) \in \beta(\alpha(v))$.

Let $u \in V(H_\ell)$. We will construct a (partial) injection $\gamma : V(G_\ell^+) \to V(G_m^+)$ such that if $uv \in R(G_\ell^+)$, then $\alpha(u)\gamma(v) \in R(G_m^+)$. Since $\alpha(u)$ has red degree at most t in G_m^+, this will prove that u has red degree at most t in G_ℓ^+. Let $v \in V(G_\ell^+)$ be a red neighbor of u in G_ℓ^+. There are two cases to be considered:

1. If $v \in V(H_\ell)$, then $\alpha(u)\alpha(v) \in R(H'_m)$, using the fact that $\beta(u), \beta(v) \subseteq V(H)$, and we set $\gamma(v) := \alpha(v)$.
2. If $v \notin V(H_\ell)$, then $v \in S_j^H$ by construction of C_j^+. Let $v_0 \in \beta(v)$. By Observation 29, $\beta(v)$ is a subset of an equivalence class of $\sim_H$. Hence, there are $u_0, u_1 \in \beta(u)$ such that $u_0v_0 \in E(G)$ but $u_1v_0 \notin E(G)$, and we let $\gamma(v) \in V(G_m^+)$ be the unique vertex such that $v_0 \in \beta(\gamma(v)) \subseteq \beta(v)$.

Now we only need to show that a vertex $v \in S_j^H$ has red degree at most $2t$ in G_ℓ^+ (no other vertex is affected by contractions among descendants of H). Let $K \subseteq V(H_\ell)$ be the set of red neighbors of v in H_ℓ (in G_ℓ^+). By Observation 29, some (actually, each) ancestor $v_0 \in V(G_m^+)$ of v has among its red neighbors all vertices of $\alpha(K)$ in G_m^+. Since the red degree of v_0 is at most t in G_m^+ and α is a bijection, we obtain that $|K| \leq t$. Hence, v has at most t red neighbors in H_ℓ (in G_ℓ^+). All other red neighbors of v in G_ℓ^+ are its red neighbors also in G_{i-1}^+ (which has maximum red degree at most t), and so v has indeed red degree at most $2t$ in G_ℓ^+. ◁

▷ Claim 34. C_k^+ has width at most $2t$.

Proof of the Claim. Let $\ell \in [j, k]$, let H_ℓ, H'_ℓ be subtrigraphs of G_ℓ^+ induced by the descendants of H and H', respectively, let $H_\ell^+ := H_\ell \cup H'_\ell$, and let $\alpha_j \colon V(H_j) \to V(H'_j)$ be the bijection defined in the construction of C_k^+. We need to show that the maximum red degree in G_ℓ^+ is at most $2t$.

First, let $v \in V(G_\ell^+ - H_\ell^+)$. By construction of C_k^+, we know that $v \in V(G_j^+)$. Suppose that v has higher red degree in G_ℓ^+ than in G_j^+. This can happen only if a black edge $uv \in E(G_j^+)$ becomes red because of a contraction involving u. However, the only contractions happening in C_k^+ are between u and $\alpha_j(u)$ for some $u \in V(H_j)$, and $uv \in E(G_j^+)$ if and only if $\alpha_j(u)v \in E(G_j^+)$, by definition of α_j. Hence, the red degree of v in G_ℓ^+ is at most its red degree in G_j^+, and that is at most $2t$ by Claim 33.

Second, we need to show that each $u \in V(H_\ell^+)$ has red degree at most $2t$ in G_ℓ^+. Observe that H'_ℓ contains no black edges because each vertex $u \in V(H'_\ell)$ is a descendant of both H' and H''. Hence, a vertex $u \in V(H'_\ell) \setminus V(H_\ell)$ has red degree at most t in G_ℓ^+ because it cannot have higher red degree in G_ℓ^+ than in G_j^+. Conversely, let $u \in V(H_\ell)$ and let d be the degree of the ancestor $u_0 \in V(H_j)$ of u in H_j. Observe that u has degree at most $2d$ in H_ℓ^+: for each neighbor $v_0 \in V(H_j)$ of u_0 in H_j, u can have two neighbors in H_ℓ^+, namely v_0 and $\alpha_j(v_0)$; this happens when u_0 has been contracted with $\alpha_j(u_0)$ into u but no neighbor $v_0 \in V(H_j)$ of u_0 has been contracted with $\alpha_j(v_0)$. Moreover, u_0 and $\alpha_j(u_0)$ have exactly the same red neighbors in S_j^H (by definition of α_j). Hence, the red degree of u in G_ℓ^+ has increased by at most $d \leq t$, compared to the red degree of u_0 in G_j^+, and so u has at most $t + d \leq 2t$ red neighbors, which concludes the proof. ◁

Since H'_j contains no black edges (each of its vertices is a descendant of both H' and H''), the contraction of H_j and H'_j creates no new red edge (using also the fact that H_j and H'_j are attached to S_j^H in the same way). Hence, we obtain that $G_k^+ \cong G_j^+ - H_j \cong G'_{i-1}$, and we can indeed define the rest of C^+ to be the suffix of C' starting with G'_i. This suffix has width at most $2t$, since C' has progressive width $(t \to_i 2t)$. ◀

Now we are finally ready to prove Lemma 22. This is the only remaining part of this section because we have already shown how Lemma 22 implies Theorem 23, see Subsection 5.1.

Proof of Lemma 22. The idea of the proof is to iteratively apply Lemma 32 to all the graphs in $\mathcal{C}$ not present in the reduced graph G'. However, this requires some care, as applying the lemma in the wrong order might fail to ensure the precondition on the progressive-width. In order to prove this lemma, we will consider the following key claim:

▷ Claim 35. Given G^*, C^*, and $\mathcal{L}^*$ satisfying the following properties, we can construct in polynomial time a contraction sequence C of width at most $2t$ for G.

1. G^* is a $\mathcal{C}$-respecting graph;
2. $\mathcal{L}^*$ is a list of pairs (graph H, integer δ), such that the integer value is non-increasing;
3. Each pair (H, δ) in $\mathcal{L}^*$ satisfies all of the following: (i.) $H \in \mathcal{C}$, and H appears only once in $\mathcal{L}^*$, (ii.) $H \nsubseteq G^*$, (iii.) $[H]_\sim$ is large in G^*, (iv.) G_δ^* is C^*-safe for H;
4. C^* is a contraction sequence for G^* of width at most $2t$, and if (H_0, δ_0) is the first pair in $\mathcal{L}^*$, then C^* has progressive width $(t \to_{\delta_0+1} 2t)$;
5. $V(G^*) \cup \bigcup_{(H,\delta)\in\mathcal{L}^*} V(H) = V(G)$.

Proof of Claim 35. We proceed by induction on the length of $\mathcal{L}^*$. The base case is trivial: if $\mathcal{L}^*$ is the empty list, the conditions 1. and 5. ensure that $G^* = G$, and 4. ensures that C^* has width $2t$.

Now let us suppose that the claim is true for any list of length i, for some $i \geq 0$. Consider G^*, C^*, $\mathcal{L}^*$ satisfying the hypothesis such that $\mathcal{L}^*$ contains $i+1$ elements, the first of which being (H_0, δ_0). We can apply Lemma 32 to G^*, C^* and H_0 since the points 1., 3. and 4. are exactly the preconditions of the lemma, and we obtain in polynomial time a contraction sequence C^+ of progressive width $(t \to_{\delta_0+1} 2t)$ for $G^+ = G[V(G^*) \cup V(H_0)]$.

Now let us consider $\mathcal{L}^+$ the suffix of $\mathcal{L}^*$ of length i – i.e., we only remove (H_0, δ_0) – and prove that G^+, C^+, and $\mathcal{L}^+$ satisfy the requirements to apply the induction hypothesis.

The first obvious point is that the length of $\mathcal{L}^+$ is i. Since G^* is $\mathcal{C}$-respecting and $H \in \mathcal{C}$, we obtain that G^+ is $\mathcal{C}$-respecting, i.e., it satisfies 1. We can easily verify 5.:

$$V(G^+) \cup \bigcup_{(H,\delta)\in\mathcal{L}^+} V(H) = V(G^*) \cup V(H_0) \bigcup_{(H,\delta)\in\mathcal{L}^+} V(H) = V(G^*) \cup \bigcup_{(H,\delta)\in\mathcal{L}^*} V(H) = V(G).$$

As a suffix of $\mathcal{L}^*$, $\mathcal{L}^+$ satisfies 2., and the first three requirements in 3. are also trivially satisfied. To prove the 3.iv., it is necessary to observe two things. First, observe that for each pair $(H, \delta) \in \mathcal{L}^+$, it holds that $G_\delta^+ = G_\delta^* \uparrow G^+$ since $\delta \leq \delta_0$, by Lemma 32 (the "moreover" part). Second, observe that there is no red edge in G_δ^+ that is not already present in G_δ^*: indeed, any such red edge would be incident to H_0 by construction of G_δ^+, and its existence would contradict the definition of δ_0, i.e., the C^*-safeness for H_0 of $G_{\delta_0}^*$. Hence we conclude that for every $(H, \delta) \in \mathcal{L}^+$, it holds that G_δ^+ is C^+-safe for H.

The last item to check, requirement 4., is easily handled: we know that C^+ has progressive width $(t \to_{\delta_0+1} 2t)$ by Lemma 32, and for all $(H, \delta) \in \mathcal{L}^+$, it holds that $\delta \leq \delta_0$ by 2., i.e., by the monotony of $\mathcal{L}^*$ in the second component.

Using the induction hypothesis, we can now create in polynomial time a contraction sequence C of width at most $2t$ for G. The total running time is polynomial, hence the claim is proven. $\lhd$

To finish the proof of Lemma 22, we only need to construct the initial list $\mathcal{L}'$ for G'. For each graph $H \in \mathcal{C}$ such that $H \nsubseteq G'$, let $\delta(H)$ be the index of the last C'-safe trigraph for H, whose existence is ensured by Lemma 30. Let $\mathcal{L}'$ be the list of pairs $(H, \delta(H))$, ordered by non-increasing values of $\delta(H)$, and recall that C' is given. It is easy to see that the requirements on G', C' and $\mathcal{L}'$ are satisfied – either by definition or by construction – to apply Claim 35: we obtain in polynomial time a contraction sequence C of width at most $2t$ for G, and since the creation of $\mathcal{L}'$ can be achieved in polynomial time, we have proven the lemma. ◀

6 Concluding Remarks

While we believe that the results presented here provide an important contribution to the state of the art in the area of computing twin-width, many prominent questions still remain unanswered. Apart from the "grand prize" – resolving the parameterized approximability of twin-width when the runtime parameter is twin-width itself – future research may focus on finding fixed-parameter algorithms that compute optimal or near-optimal contraction sequences under less restrictive runtime parameters than those considered in this article.

More specifically, the problem remains entirely open when parameterized by treewidth and treedepth, and resolving this may require new insights into the structural properties of optimal contraction sequences and lead to tighter bounds on the twin-width of well-structured graphs. For treedepth in particular, we suspect that combining the ideas presented in Section 5 with the *iterative pruning* approach typically used for treedepth-based algorithms [26, 28, 8] may be an enticing direction to pursue; however, we note that such a combination does not seem straightforward.

References

1 Jungho Ahn, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. Bounds for the twin-width of graphs. *SIAM Journal on Discrete Mathematics*, 36(3):2352–2366, 2022. doi:10.1137/21M1452834.

2 Jakub Balabán, Robert Ganian, and Mathis Rocton. Computing twin-width parameterized by the feedback edge number. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, March 12-14, 2024, Clermont-Ferrand, France*, volume 289 of *LIPIcs*, pages 7:1–7:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.STACS.2024.7.

3 Jakub Balabán and Petr Hlinený. Twin-width is linear in the poset width. In Petr A. Golovach and Meirav Zehavi, editors, *16th International Symposium on Parameterized and Exact Computation, IPEC 2021, September 8-10, 2021, Lisbon, Portugal*, volume 214 of *LIPIcs*, pages 6:1–6:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.IPEC.2021.6.

4 Jakub Balabán, Petr Hlinený, and Jan Jedelský. Twin-width and transductions of proper k-mixed-thin graphs. In Michael A. Bekos and Michael Kaufmann, editors, *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers*, volume 13453 of *Lecture Notes in Computer Science*, pages 43–55. Springer, 2022. doi:10.1007/978-3-031-15914-5_4.

5 Michael J. Bannister, Sergio Cabello, and David Eppstein. Parameterized complexity of 1-planarity. *J. Graph Algorithms Appl.*, 22(1):23–49, 2018. doi:10.7155/jgaa.00457.

6 Pierre Bergé, Édouard Bonnet, and Hugues Déprés. Deciding twin-width at most 4 is np-complete. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 18:1–18:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.18.

7 Benjamin Bergougnoux, Eduard Eiben, Robert Ganian, Sebastian Ordyniak, and M. S. Ramanujan. Towards a polynomial kernel for directed feedback vertex set. *Algorithmica*, 83(5):1201–1221, 2021. doi:10.1007/s00453-020-00777-5.

8 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for queue layouts. *J. Graph Algorithms Appl.*, 26(3):335–352, 2022. doi:10.7155/JGAA.00597.

9 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Preprocessing for treewidth: A combinatorial analysis through kernelization. *SIAM J. Discret. Math.*, 27(4):2108–2142, 2013. doi:10.1137/120903518.

10 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width II: small classes. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1977–1996. SIAM, 2021. doi:10.1137/1.9781611976465.118.

11 Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width III: max independent set, min dominating set, and coloring. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.35.

12 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Torunczyk. Twin-width IV: ordered graphs and matrices. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 924–937. ACM, 2022. doi:10.1145/3519935.3520037.

13 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, and Stéphan Thomassé. Twin-width V: linear minors, modular counting, and matrix multiplication. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, March 7-9, 2023, Hamburg, Germany*, volume 254 of *LIPIcs*, pages 15:1–15:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.STACS.2023.15.

14 Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, and Stéphan Thomassé. Twin-width VI: the lens of contraction sequences. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 1036–1056. SIAM, 2022. doi:10.1137/1.9781611977073.45.

15 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022. doi:10.1145/3486655.

16 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015. doi:10.1007/978-3-319-21275-3.

17 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics.* Springer, 2012.

18 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

19 Pål Grønås Drange, Markus S. Dregi, and Pim van 't Hof. On the computational complexity of vertex integrity and component order connectivity. *Algorithmica*, 76(4):1181–1202, 2016. doi:10.1007/S00453-016-0127-X.

20 Pavel Dvorák, Eduard Eiben, Robert Ganian, Dusan Knop, and Sebastian Ordyniak. The complexity landscape of decompositional parameters for ILP: programs with few global variables and constraints. *Artif. Intell.*, 300:103561, 2021. doi:10.1016/J.ARTINT.2021.103561.

21 Eduard Eiben, Robert Ganian, Thekla Hamm, Lars Jaffke, and O-joung Kwon. A unifying framework for characterizing and computing width measures. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 63:1–63:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.63.

22 David Eppstein. The widths of strict outerconfluent graphs. *CoRR*, abs/2308.03967, 2023. arXiv:2308.03967.

23 Johannes Klaus Fichte, Robert Ganian, Markus Hecher, Friedrich Slivovsky, and Sebastian Ordyniak. Structure-aware lower bounds and broadening the horizon of tractability for QBF. In *LICS*, pages 1–14, 2023. doi:10.1109/LICS56636.2023.10175675.

24 Robert Ganian, Fabian Klute, and Sebastian Ordyniak. On structural parameterizations of the bounded-degree vertex deletion problem. *Algorithmica*, 83(1):297–336, 2021. doi:10.1007/S00453-020-00758-8.

25 Robert Ganian and Viktoriia Korchemna. The complexity of bayesian network learning: Revisiting the superstructure. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 430–442, 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/040a99f23e8960763e680041c601acab-Abstract.html.

26 Robert Ganian and Sebastian Ordyniak. The complexity landscape of decompositional parameters for ILP. *Artif. Intell.*, 257:61–71, 2018. doi:10.1016/J.ARTINT.2017.12.006.

27 Robert Ganian and Sebastian Ordyniak. The power of cut-based parameters for computing edge-disjoint paths. *Algorithmica*, 83(2):726–752, 2021. doi:10.1007/s00453-020-00772-w.

28 Robert Ganian, Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Fixed-parameter tractability of dependency QBF with structural parameters. In Diego Calvanese, Esra Erdem, and Michael Thielscher, editors, *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pages 392–402, 2020. doi:10.24963/KR.2020/40.

29 Tatsuya Gima, Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, and Yota Otachi. Exploring the gap between treedepth and vertex cover through vertex integrity. *Theor. Comput. Sci.*, 918:60–76, 2022. doi:10.1016/J.TCS.2022.03.021.

30 Tatsuya Gima and Yota Otachi. Extended MSO model checking via small vertex integrity. *Algorithmica*, 86(1):147–170, 2024. doi:10.1007/S00453-023-01161-9.

31 Tesshu Hanaka, Michael Lampis, Manolis Vasilakis, and Kanae Yoshiwatari. Parameterized vertex integrity revisited. *CoRR*, abs/2402.09971, 2024. doi:10.48550/arXiv.2402.09971.

32 Hugo Jacob and Marcin Pilipczuk. Bounding twin-width for bounded-treewidth graphs, planar graphs, and bipartite graphs. In Michael A. Bekos and Michael Kaufmann, editors, *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers*, volume 13453 of *Lecture Notes in Computer Science*, pages 287–299. Springer, 2022. doi:10.1007/978-3-031-15914-5_21.

33 Yasuaki Kobayashi and Hisao Tamaki. Treedepth parameterized by vertex cover number. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63 of *LIPIcs*, pages 18:1–18:11. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.IPEC.2016.18.

34 Michael Lampis and Valia Mitsou. Fine-grained meta-theorems for vertex integrity. In Hee-Kap Ahn and Kunihiko Sadakane, editors, *32nd International Symposium on Algorithms and Computation, ISAAC 2021, December 6-8, 2021, Fukuoka, Japan*, volume 212 of *LIPIcs*, pages 34:1–34:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.ISAAC.2021.34`.

35 Sang-il Oum. Rank-width and vertex-minors. *J. Comb. Theory, Ser. B*, 95(1):79–100, 2005. `doi:10.1016/j.jctb.2005.03.003`.

36 Neil Robertson and Paul D. Seymour. Graph minors. i. excluding a forest. *J. Comb. Theory B*, 35(1):39–61, 1983. `doi:10.1016/0095-8956(83)90079-5`.

37 Johannes Uhlmann and Mathias Weller. Two-layer planarization parameterized by feedback edge set. *Theor. Comput. Sci.*, 494:99–111, 2013. `doi:10.1016/j.tcs.2013.01.029`.

38 Martijn van Ee. Some notes on bounded starwidth graphs. *Inf. Process. Lett.*, 125:9–14, 2017. `doi:10.1016/J.IPL.2017.04.011`.