

Twin-Width One

Jungho Ahn ✉ 🏠 

Korea Institute for Advanced Study (KIAS), Seoul, South Korea

Hugo Jacob ✉ 

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Noleen Köhler ✉ 

University of Leeds, UK

Christophe Paul ✉ 

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Amadeus Reinald ✉ 

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Sebastian Wiederrecht ✉ 

School of Computing, KAIST, Daejeon, South Korea

Abstract

We investigate the structure of graphs of twin-width at most 1, and obtain the following results:

- Graphs of twin-width at most 1 are permutation graphs. In particular they have an intersection model and a linear structure.
- There is always a 1-contraction sequence closely following a given permutation diagram.
- Based on a recursive decomposition theorem, we obtain a simple algorithm running in linear time that produces a 1-contraction sequence of a graph, or guarantees that it has twin-width more than 1.
- We characterise distance-hereditary graphs based on their twin-width and deduce a linear time algorithm to compute optimal sequences on this class of graphs.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Twin-width, Hereditary graph classes, Intersection model

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.6

Related Version *Full Version:* [arxiv:2501.00991](https://arxiv.org/abs/2501.00991)

Funding *Jungho Ahn:* Supported by the KIAS Individual Grant (CG095301) at Korea Institute for Advanced Study.

Hugo Jacob: Supported by the ANR project GODASse ANR-24-CE48-4377.

Christophe Paul: Supported by the ANR project GODASse ANR-24-CE48-4377 and the ANR-DFG project UTMA ANR-20-CE92-0027.

Amadeus Reinald: Supported by the ANR project DIGRAPHS ANR-19-CE48-0013.

Acknowledgements The authors wish to thank the organisers of the 1st Workshop on Twin-width, which was partially financed by the grant ANR ESIGMA (ANR-17-CE23-0010) of the French National Research Agency. The second author wishes to thank Paul Bastide and Carla Groenland for interesting initial discussions on the topic of this paper.

1 Introduction

Twin-width is a graph invariant introduced by Bonnet, Kim, Thomassé, and Watrigant [11] as a generalisation of a parameter on permutations introduced by Guillemot and Marx [21]. The main result of the seminal paper [11] is an FPT algorithm for FO model-checking on graphs of bounded twin-width, when given a *contraction sequence* certifying their width. Graphs of



© Jungho Ahn, Hugo Jacob, Noleen Köhler, Christophe Paul, Amadeus Reinald, and Sebastian Wiederrecht;

licensed under Creative Commons License CC-BY 4.0

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).

Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thăng;

Article No. 6; pp. 6:1–6:19



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



bounded twin-width capture a wide variety of classes such as bounded rank-width and proper minor-closed classes, unifying many results on tractable FO model-checking. Furthermore, it gives an exact dichotomy between tractability and intractability of FO model-checking for hereditary classes of ordered structures [8] and of tournaments [19].

While these results are sufficient to consider twin-width an important graph parameter, we are still lacking an FPT algorithm for computing “good” contraction sequences, i.e. contraction sequences of width bounded by a function of the twin-width. Such an algorithm is required for FO model-checking to be FPT on bounded twin-width classes. Regarding exact algorithms for twin-width, it is known that computing twin-width is hard even for constant values: distinguishing graphs of twin-width 4 from graphs of twin-width 5 is NP-hard [5]. Still, a polynomial-time algorithm for recognizing graphs of twin-width 1 was given in [10], where the worst-case time complexity is not explicitly given but corresponds to a polynomial of degree 4 or 5 depending on implementation. Meanwhile, the hardness of recognizing twin-width 2 and 3 graphs remains open. Nevertheless, it is known that sparse graphs of twin-width 2 have bounded treewidth [7]. As for approximation, there is an algorithm to compute contraction sequences of approximate width for ordered structures [8]. It is still wide open whether there is an XP, let alone FPT, approximation algorithm in the general case. Regarding parameterized algorithms, some results are known with parameters that are still quite far from twin-width [3, 4]. Interestingly, the more general parameter flip-width introduced by Torunczyk [25] has an XP approximation algorithm but the status of FO model-checking for this parameter is unknown.

In this paper, we are interested in the structure of graphs of twin-width 1, and in the complexity of their recognition. Let us briefly recall the definition of k -contraction sequences, which are witnesses for twin-width at most k . Given a graph G , a contraction sequence of G starts with G , and consists in a sequence of identifications of pairs of vertices (contractions) recording “neighbourhood errors” as *red edges*, ending with a graph on a single vertex. Specifically, at each step, we create red edges from the contracted vertex to vertices that were not homogeneous to the pair. That is, vertices which were not both adjacent or non-adjacent to the pair. In particular, red edges remain red. Then, a k -contraction sequence is one where at each step the vertices have most k incident red edges.

Towards understanding graphs of twin-width one, it will be useful to look at the structure and behaviour of classes of graphs which are related. There is a rich literature on hereditary classes of graphs related to perfect graphs. It follows from the following observation and the strong perfect graph theorem [13] that twin-width 1 graphs are perfect graphs. It is then natural to wonder how they compare to other subclasses of perfect graphs.

► **Observation 1** (See [1, Lemma 2.3]). *Every cycle of length at least 5 has twin-width 2. This also implies that complements of cycles of length at least 5 have twin-width 2.*

A natural way of understanding classes of twin-width at most k is to interpret them as a generalization of cographs. Indeed, cographs are exactly the graphs which admit a sequence of twin identifications ending in a single vertex, that is, graphs of twin-width 0. Then, graphs of twin-width k correspond to relaxing the twin condition to allow for at most k “twin errors”.

We now compare twin-width 1 graphs with other generalisations of cographs. Cographs are the graphs of clique-width at most 2. In this direction, cographs are generalised by the class of distance-hereditary graphs which are themselves closely related to graphs of clique-width at most 3 [14]. Distance-hereditary (DH) graphs are not closed by complementation, and have a tree-like structure. They also have a characterisation by a sequential elimination of twins and pendant vertices. Cographs are permutation graphs, which is exactly the complementation-closed class of graphs whose edges can be transitively oriented [17]. Permutation graphs are

asteroidal triple-free (AT-free), as such, they are dominated by a path [15]. A caterpillar is an AT-free tree, the following result suggests a relation between AT-free graphs and twin-width 1.

► **Lemma 2** (Ahn, Hendrey, Kim, and Oum [2, Lemma 6.6]). *For a tree T , $\text{tw}(T) \leq 1$ if and only if T is a caterpillar.*

Cographs, distance-hereditary graphs, and permutation graphs can all be recognised in linear time. It is only natural to expect that the closely related class of twin-width 1 graphs can also be recognised efficiently.

Previous results on graphs of twin-width at most 1

The recognition algorithm for twin-width one given in [10] makes use of the relation between modules and twin-width, by observing that the twin-width of a graph can be determined by considering independently the subgraphs induced by modules (see Lemma 5). It is straightforward to deduce that the twin-width of a graph is the maximal twin-width over all prime nodes of its modular decomposition (defined in Section 2). Since this decomposition may be computed in linear time [24], one then only needs to recognise prime graphs of twin-width one. In their paper [10], the recognition is done by branching on valid sequences that have at most one red edge at any intermediate step of the sequence, after the observation that such sequences always exist. Furthermore, when the trigraph has a red edge, the only possible contractions in such a sequence involve at least one vertex of the red edge. Essentially, the complexity of their algorithm stems from the need to branch over all possible first contractions, to greedily simulate a contraction sequence for each possible first contraction, and the lack of an efficient way to detect eventual twins to be contracted at each step. In particular, a more explicit understanding of the structure of graphs of twin-width at most 1 seems to be required to avoid the enumeration of all pairs of possible first contractions.

Our results

We investigate the structure of graphs of twin-width at most 1, and uncover that it defines a relatively well-behaved hereditary class of graphs, which fits surprisingly well in the landscape of classical hereditary classes. Graphs of twin-width at most 1 were already known to have some form of linear structure, due to a bound on their linear rankwidth if they are prime (see Lemma 7 and [9]). We show that they are in fact contained in the class of permutation graphs, thus they admit an intersection model called “permutation diagram”, making it easier to reason on their structure. Furthermore, we show that there is a close relationship between permutation diagrams and 1-contraction sequences. We use these structural results to obtain a recursive decomposition theorem for prime graphs of twin-width at most 1. We then use it to devise a simple linear time algorithm to compute a 1-contraction sequence, or conclude that the graph has twin-width more than 1. The algorithm starts by computing a permutation diagram and a modular decomposition which can be done in time $O(n + m)$, where n is the number of vertices and m is the number of edges, and then it checks that the diagram respects the decomposition theorem in time $O(n)$.

Regarding the challenge of avoiding the enumeration of possible first contractions, we can significantly reduce the number of candidates by using the permutation diagram and related properties of contraction sequences. We consider a slightly different scheme to avoid this challenge. Instead of guessing the first contraction, we guess the vertex that will be contracted last. It turns out that, in well-behaved contraction sequences, such a vertex

holds a special position in permutation diagrams, reducing the number of candidates to 4. However, we do not exclude the possibility that the scheme of the previous algorithm can be implemented in linear time with some further arguments.

Interestingly, our structural analysis via the permutation diagrams allows us to avoid a characterisation by forbidden induced subgraphs. Moreover, we prove most structural tools inductively and in a unified way instead of proving the observations required for our algorithm separately.

In the full version, we also present some intermediate results on distance-hereditary graphs and split decompositions that turned out to be unnecessary for the recognition algorithm thanks to the use of permutation diagrams. We show that twin-width 1 trigraphs with a pendant red edge are distance-hereditary. We also show that distance-hereditary graphs have twin-width at most 2. Combining these results, we can compute an optimal contraction sequence for distance-hereditary graphs in linear time. We also obtain a simple inductive proof that all distance-hereditary AT-free graphs are permutation graphs, and a characterisation of the twin-width of a distance-hereditary graph by the structure of its split decomposition. This is a generalisation of the characterisation of twin-width 1 trees [2] mentioned above.

Distance-hereditary graphs are also related to the following infinite family of permutation graphs of twin-width 2: linking two obstructions to distance-hereditary graphs (domino, house, or gem) by a path of arbitrary length. In particular, this class of examples shows that a linear time algorithm for the recognition of twin-width 1 graphs cannot be deduced from the linear algorithm to find patterns in a permutation of Guillemot and Marx [21].

We also observe that bipartite graphs of twin-width 1 constitute a well-behaved subclass. Indeed, in their 1-contraction sequences, all trigraphs are bipartite. This is because there are no induced odd cycles of length more than 3, and triangles have at most one red edge from which we can deduce an original edge and complete it in a triangle of the starting graph (this is a particular case of Observation 14).

Perspectives

One may hope that understanding the structure of graphs of twin-width 1 well enough can guide a subsequent study of graphs of twin-width 2. We expect that this hereditary class can be characterised by a tree-like decomposition with a very constrained local structure. Our results also point toward the fact that split decompositions could be a convenient tool for the analysis of the structure of twin-width 2 graphs.

In another direction, one may wonder if it is possible to compute other values of the twin-width efficiently for permutation graphs. An FPT approximation algorithm is already known in this setting [11], although one may try to find tighter approximation bounds.

Organisation

We organise this paper as follows. In Section 2, we present some terminology and useful known results on twin-width, and on permutation graphs. In Section 3, we show that every graph of twin-width at most 1 is a permutation graph. In Section 4, we present a linear-time algorithm that recognises graphs of twin-width at most 1.

2 Preliminaries

In this paper, all graphs are finite and simple. For an integer i , we denote by $[i]$ the set of positive integers at most i . Note that if $i \leq 0$, then $[i]$ is an empty set. A subset $I \subseteq [i]$ is an *interval* of $[i]$ if there are integers $i_1, i_2 \in [i]$ such that $I = \{j : i_1 \leq j \leq i_2\}$.

Let G be a graph. We denote the complement graph by \overline{G} . For a vertex v of G , we denote by $N_G(v)$ the set of neighbours of v , and let $N_G[v] := N_G(v) \cup \{v\}$. Then, $\overline{N}_G(v) = V(G) \setminus N[v]$. For a set $X \subseteq V(G)$, let $N_G[X] := \bigcup_{v \in X} N_G[v]$ and let $N_G(X) := N_G[X] \setminus X$. Distinct vertices v and w of G are *twins in G* if $N_G(v) \setminus \{w\} = N_G(w) \setminus \{v\}$. For a set $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of G induced by X . A *dominating set* of G is a set $D \subseteq V(G)$ such that $N_G[D] = V(G)$. A *dominating path* of G is a path whose vertex set is a dominating set of G . Three vertices form an *asteroidal triple* if each pair of vertices is connected by a path that does not dominate the third vertex. A graph is *asteroidal triple-free* (AT-free) if it contains no asteroidal triple. An *independent set* of a graph G is a set $I \subseteq V(G)$ such that G has no edge between two vertices in I .

A *module* of a graph G is a set $M \subseteq V(G)$ such that for every vertex $v \in V(G) \setminus M$, v is either adjacent to every vertex in M , or nonadjacent to every vertex in M . If M is not a module, we call *splitter* of M a vertex $v \in V(G) \setminus M$, that has both a neighbour and a non-neighbour in M . Note that M is a module exactly if it has no splitter. A module is *trivial* if it either has size at most 1, or is equal to $V(G)$. A graph is *prime with respect to modules*, or simply *prime*, if all of its modules are trivial. A *modular partition* of a graph G is a partition of $V(G)$ where each part is a module of G . For a modular partition $\mathcal{M} = (M_1, \dots, M_\ell)$ of a graph G , we denote by G/\mathcal{M} , the subgraph of G induced by a set containing exactly one vertex from each M_i . A module M is *strong* if it does not overlap with any other module, i.e. for every module M' , we have $M \subseteq M'$, or $M' \subseteq M$, or $M \cap M' = \emptyset$.

The *modular decomposition* of a graph G is a tree T corresponding to the recursive modular partition by the maximal strong modules. The leaves of the tree are vertices of G , and each internal node n has an associated quotient graph $Q(n)$ which encodes the adjacency relation between vertices of G introduced in subtrees below n . In particular, quotient graphs are induced subgraphs of G , and the modular decomposition of an induced subgraph of G can easily be deduced from the modular decomposition of G . There are three types of internal nodes: *series*, *parallel*, and *prime*, corresponding to cliques, independent sets and prime graphs (with respect to modules). Prime graphs are graphs whose modules are all trivial i.e. singletons and the complete vertex set. Series and parallel nodes are called *degenerate*, and two degenerate nodes of the same type may not be adjacent in T . The reason they are called degenerate is because any subset of vertices of a clique or an independent set is a module. Cographs are exactly the graphs whose modular decompositions have no prime nodes, their modular decomposition is usually called a cotree.

2.1 Permutation graphs

A graph G on n vertices is a *permutation graph* if there are two linear orderings $\sigma : V(G) \rightarrow [n]$ and $\tau : V(G) \rightarrow [n]$ such that two vertices u and v are adjacent in G if and only if $\{u, v\}$ is an inversion of $\sigma^{-1} \circ \tau$. We remark that $\sigma^{-1} \circ \tau$ and its inverse permutation $\tau^{-1} \circ \sigma$ have the same set of inversions. A *permutation diagram* of G with respect to σ and τ is a drawing of n line segments between two parallel lines ℓ_1 and ℓ_2 such that each of ℓ_1 and ℓ_2 has n distinct points, and for each $v \in V(G)$, there is a line segment between $\sigma(v)$ -th point of ℓ_1 and $\tau(v)$ -th point of ℓ_2 . Note that two vertices are adjacent in G if and only if their corresponding line segments intersect each other in the permutation diagram.

We denote by $G[\sigma, \tau]$ the graph on V where uv is an edge if and only if $\{u, v\}$ is an inversion of $\sigma^{-1} \circ \tau$. As pointed earlier, we have $G[\sigma, \tau] = G[\tau, \sigma]$. If $G = G[\sigma, \tau]$, we call (σ, τ) a *realiser* of G (as a permutation graph).

For σ an ordering of V , we define intervals of V as follows: for $\{i, i+1, \dots, j-1, j\}$ an interval of $[n]$, we have an interval of V for σ $\{\sigma^{-1}(i) = u, \dots, \sigma^{-1}(j) = v\}$ that we denote by $[u, v]_\sigma$. We extend the notation to open intervals similarly.

We say that $I \subseteq V$ is an *interval* of realiser (σ, τ) if it is an interval for σ or for τ . We also say that $u, v \in V(G_i)$ are *consecutive* for (σ, τ) if $\{u, v\}$ is an interval of (σ, τ) . Similarly, two intervals are *consecutive* if their union is an interval. We say that I is a *common interval* of realiser (σ, τ) if it is an interval for σ and for τ . A vertex v of G is *extremal* for realiser (σ, τ) (or equivalently the corresponding permutation diagram) if it is the first or last vertex of σ or τ .

It is known that for a prime permutation graph, its permutation diagram (or equivalently its realiser) is unique up to symmetry [18, 20]. More generally, the modular decomposition encodes all possible realisers. Furthermore, we have the following known observation which will be very convenient for our analysis.

► **Observation 3.** *If M is a common interval of a realiser of G , then M is a module of G . Conversely, if M is a strong module of G , then it is a common interval of all realisers.*

It will also be useful to note that extremal vertices of a prime permutation graph do not depend on the realiser. See [6, 12, 16, 22] for further details on this.

Moreover, for a permutation diagram of G with respect to linear orderings σ and τ of $V(G)$, if we reverse one of σ and τ , then we obtain a permutation diagram of \overline{G} . In other words, \overline{G} has a permutation diagram with respect to σ and $n+1-\tau$, or to $n+1-\sigma$ and τ .

Thus, a graph G is a permutation graph if and only if \overline{G} is a permutation graph.

2.2 Trigraphs and twin-width

A *trigraph* is a triple $H = (V(H), B(H), R(H))$ where $B(H)$ and $R(H)$ are disjoint sets of unordered pairs of $V(H)$. We denote by $E(H)$ the union of $B(H)$ and $R(H)$. The *edges* of H are the elements in $E(H)$. The *black edges* of H are the elements of $B(H)$, and the *red edges* of H are the elements of $R(H)$. We identify a graph $G = (V, E)$ with a trigraph (V, E, \emptyset) .

The *underlying graph* of a trigraph H is the graph $(V(H), E(H))$. We identify H with its underlying graph when we use standard graph-theoretic terms and notations. A *black neighbour* of v is a vertex u with $uv \in B(H)$ and a *red neighbour* of v is a vertex w with $vw \in R(H)$. The *red degree* of v is the number of red neighbours of v . For an integer d , a *d-trigraph* is a trigraph with maximum red degree at most d . For a set $X \subseteq V(H)$, we denote by $H - X$ the trigraph obtained from H by removing all vertices in X . If $X = \{v\}$, then we write $H - v$ for $H - X$.

For a trigraph H and distinct vertices u and v of H , we denote by $H/\{u, v\}$ the trigraph H' obtained from $H - \{u, v\}$ by adding a new vertex x such that for every vertex $y \in V(H) \setminus \{u, v\}$, the following hold:

- if y is a common black neighbour of u and v , then $xy \in B(H')$,
- if y is adjacent to none of u and v , then $xy \notin E(H')$, and
- otherwise, $xy \in R(H')$.

We say that H' is obtained by *contracting u and v* .

A *contraction sequence* of G is a sequence of trigraphs G_n, \dots, G_1 where $G_n = G$, G_1 as a single vertex, and G_{i-1} is obtained from G_i by contracting a pair of vertices. For an integer d , a contraction sequence G_n, \dots, G_1 is a *d-sequence* if for every $i \in [t]$, the maximum red-degree of G_i is at most d . The *twin-width* of G , denoted by $\text{tw}(G)$, is the minimum integer d such that there is a d -sequence of G .

The vertices of the trigraphs in the contraction sequence can be interpreted as a partition of $V(G)$, this leads to a very natural characterisation of red edges in a trigraph as pairs of part such that the relation between vertices in the two parts is not homogeneous (i.e. there is an edge and a non-edge).

We have the following observations from [11].

► **Observation 4** (Bonnet, Kim, Thomassé, and Watrigant [11]). *For a graph G , the twin-width of G is equal to that of \overline{G} , and every induced subgraph of G has twin-width at most $\text{tw}(G)$.*

► **Lemma 5** (Bonnet, Kim, Reinald, Thomassé, and Watrigant [10, Lemma 9]). *Let G be a graph and let $\mathcal{M} = (M_1, \dots, M_\ell)$ be a modular partition. Then*

$$\text{tw}(G) = \max \left\{ \text{tw}(G/\mathcal{M}), \max_{i \in [\ell]} \text{tw}(G[M_i]) \right\}.$$

We sketch the proof as this will be of importance throughout the paper.

Proof. A contraction in a module M does not create red edges incident to $V(G - M)$. Therefore, we can always first contract the induced subgraphs $G[M_i]$ in any order, and then contract G/\mathcal{M} once all subgraphs $G[M_i]$ have been contracted to single vertices. ◀

► **Corollary 6.** *The twin-width of a graph G is equal to the maximum twin-width of the quotient graphs of nodes of its modular decomposition.*

Proof. We may always contract leaf nodes of the modular decomposition via their optimal contraction sequence until the whole graph is reduced to a single vertex. This corresponds exactly to a recursive application of the above lemma with maximal strong modules. ◀

The above statements show that computing the twin-width of a graph reduces to the case of a prime graph. Thus, throughout this paper, we focus on prime graphs of twin-width at most 1. The following lemma on the structure of twin-width 1 graphs will be useful.

► **Lemma 7** (Bonnet, Kim, Reinald, Thomassé, and Watrigant [10, Lemma 31]). *Let G be a prime graph of twin-width 1 and let $G_n (= G), \dots, G_1$ be a 1-sequence of G . Then for every $i \in [n - 1] \setminus \{1\}$, the trigraph G_i has exactly one red edge.*

3 Permutation diagrams

In this section, we show that every graph of twin-width at most 1 is a permutation graph.

► **Theorem 8.** *Every graph of twin-width at most 1 is a permutation graph.*

It would be sufficient to prove that graphs of twin-width at most 1 are comparability graphs. Indeed, since twin-width is stable by complementation, this implies that graphs of twin-width at most 1 are also co-comparability graphs. We could then conclude because permutation graphs are exactly the graphs that are both comparability and co-comparability graphs [17]. To do this, one could make use of the following characterisation of comparability graphs via forbidden induced subgraphs given by Gallai [18] (see also [23]).

► **Theorem 9.** *A graph G is a comparability graph if and only if G does not contain any graph of Figure 3 as induced subgraph, and \overline{G} does not contain any graph of Figure 4 as induced subgraph.*

We instead present a constructive proof because it gives combinatorial insights on the relationship between contraction sequences and permutation diagrams that will be useful to find an efficient recognition algorithm. The trick is to decompose the graph by considering the contraction sequence backwards and to realise that vertices that are contracted together at some point in the contraction sequence should be consecutive or almost consecutive in a realiser.

The following recursive decomposition of twin-width 1 graphs is meant as a simple introduction to the technique. This decomposition will later be strengthened in Lemma 11 and Corollary 15. We abusively extend the terminology of universal vertex and isolated vertex to modules M such that replacing M by a single vertex makes it universal or isolated.

► **Lemma 10.** *The class \mathcal{G} of graphs of twin-width at most 1 satisfies the following recursive characterisation:*

$$\mathcal{G} = \left\{ G \mid \exists M \subset V(G), \begin{array}{l} G - M \in \mathcal{G}, G[M] \in \mathcal{G}, M \text{ is a module of } G, \\ M \text{ is universal or } M \text{ is isolated or} \\ G - M \text{ admits a 1-contraction sequence ending} \\ \text{with the (possibly red) edge } \{N(M), \overline{N}(M)\} \end{array} \right\}.$$

Proof. Consider a graph G such that there exists a module $M \subset V(G)$ such that $G - M \in \mathcal{G}$ and $G[M] \in \mathcal{G}$. If M is universal or isolated, $G - M$ is a module of G and, by Lemma 5, G is also a graph of \mathcal{G} . If $G - M$ admits a partial 1-contraction sequence π ending with edge $\{N(M), \overline{N}(M)\}$, then π is also a partial 1-contraction sequence of G , and can be extended to be 1-contraction sequence of G . Indeed, no red edge incident to M appears when applying π to G since pairs of contracted vertices are always either in $N(M)$ or $\overline{N}(M)$, we can then apply the 1-contraction sequence of M , and finally contract arbitrarily the resulting 3-vertex trigraph.

Conversely, consider a graph of \mathcal{G} . It admits a 1-contraction sequence with a 3-vertex trigraph G_3 . G_3 has at most one red edge ab (pick a, b arbitrarily in $V(G_3)$ if there is no red edge). We consider $v \in V(G_3) - \{a, b\}$. The set M of vertices of G that were contracted to form v is a module since there are no incident red edges in G_3 . $G - M \in \mathcal{G}$ and $G[M] \in \mathcal{G}$ since they are induced subgraphs of G . Let A and B be the sets of vertices that were contracted to form a and b , respectively. Since v has no red neighbour, it must be that $A \subseteq N(M)$ or $A \subseteq \overline{N}(M)$, similarly, $B \subseteq N(M)$ or $B \subseteq \overline{N}(M)$. We conclude that M is universal, isolated or $G - M$ admits a contraction sequence ending with the edge $\{N(M), \overline{N}(M)\}$. ◀

Given a permutation graph $G = (V, E)$ and $U \subseteq V$, we denote by $\sigma[U]$ and $\tau[U]$ the restrictions of σ and τ (re-numbered by $[[U]]$). They form a realiser of $G[U]$ as a permutation graph. We also use the notation $\sigma \cdot \tau$ to denote the concatenation of two orderings on disjoint domains (all elements of σ are before elements of τ , relative orders in σ and τ are preserved).

The following lemma implies Theorem 8.

► **Lemma 11.** *Let G be a graph of twin-width at most 1, and G_n, \dots, G_2, G_1 be a fixed 1-contraction sequence of G .*

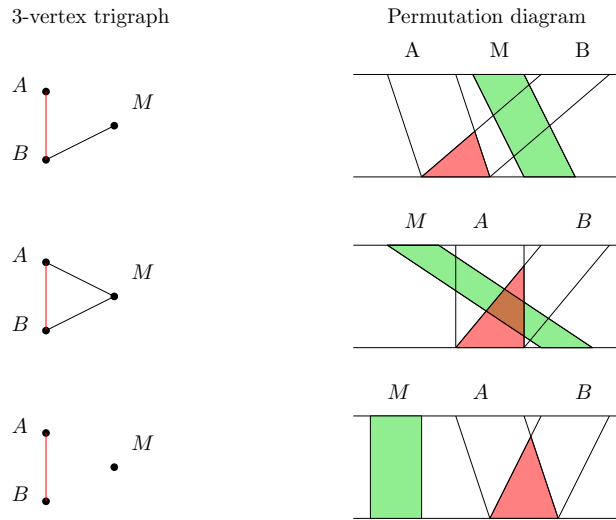
Then G is a permutation graph admitting a realiser (σ, τ) satisfying the following.

■ *For every $x \in V(G_i)$, $\{v \in V(G) \mid v \in x\}$ is an interval of (σ, τ) .*

- For every $x, y \in V(G_i)$, if $xy \in R(G_i)$, then $\{v \in V(G) \mid v \in (x \cup y)\}$ is an interval on one of the orderings $\pi \in \{\sigma, \tau\}$, and both $\{v \in V(G) \mid v \in x\}$ and $\{v \in V(G) \mid v \in y\}$ are intervals on the other ordering.

Proof. We proceed by induction on the number of vertices of the graph by leveraging the recursive characterization given in Lemma 10.

The statement is trivial for graphs on at most 3 vertices. We now consider a graph $G \in \mathcal{G}$ on more than 3 vertices and its contraction sequence $G_n, \dots, G_3, G_2, G_1$. Let M be the set of vertices of G that were contracted into a vertex of G_3 not incident to a red edge in G_3 , M is a module of G . Restricting the sequence G_n, \dots, G_1 to $G - M$ and $G[M]$, respectively, yields a 1-contraction sequence for each of them. By induction hypothesis applied with these sequences, we obtain realisers σ_1, τ_1 and σ_2, τ_2 of $G - M$ and $G[M]$, respectively. The following cases are illustrated in Figure 1. If M is universal (resp. isolated), $\sigma_1 \cdot \sigma_2, \tau_2 \cdot \tau_1$ (resp. $\sigma_1 \cdot \sigma_2, \tau_1 \cdot \tau_2$) is a realiser of G . Otherwise, the contraction sequence $G_n - M, \dots, G_3 - M$ is such that $V(G_3 - M) = \{N(M), \overline{N}(M)\}$. In particular, we know from the induction hypothesis that $B = N(M)$ and $A = \overline{N}(M)$ are intervals of either σ_1 or τ_1 , as they correspond to the sets of vertices contracted into the last two vertices on the sequence. Without loss of generality, we assume they are intervals of σ_1 . We can now produce the realiser $\sigma_1[B] \cdot \sigma_2 \cdot \sigma_1[A], \tau_2 \cdot \tau_1$. In all cases, the realiser satisfies the desired properties: it suffices to observe that by construction we do not contract vertices from $G - M$ with vertices from $G[M]$ until the last two contractions, so we must check only the last two contractions which are on trigraphs of order at most 3 hence the properties are satisfied. ◀



■ **Figure 1** The different cases of the induction constructing the realiser.

We now move on to show related properties that will be useful in further understanding permutation diagrams and their relation to contraction sequences.

In the particular case of a prime graph, we deduce the following two corollaries of Lemma 11.

- **Corollary 12.** *In a prime graph G of twin-width 1, the first contraction must involve two vertices that are consecutive for one ordering, and with exactly one vertex between them for the other ordering.*

Proof. In G_{n-1} , let x', z be the vertices incident to the red edge (there is a red edge because the graph is prime, see Lemma 7) with x' resulting of the contraction of vertices x and y of G . $X' = \{x, y\}$ is an interval for one ordering σ of the realiser. z must be a splitter of X' because it is incident to the red edge, this implies that z is between x and y in the ordering τ . ◀

Unfortunately, there can be linearly many such pairs. On the other hand, there are only 4 extremal vertices in a prime permutation graph. Hence, the following corollary is useful for obtaining a linear time recognition algorithm.

► **Corollary 13.** *In a prime graph of twin-width 1, for any 1-contraction sequence, the last vertex to become incident to the red edge is an extremal vertex of the realiser.*

Proof. Due to the graph being prime, the last vertex of G to become incident to the red edge is also a vertex of G_3 and is adjacent to exactly one other vertex of G_3 . The other two vertices of G_3 are connected because prime graphs are connected, hence G_3 is isomorphic to P_3 whose vertices are all extremal. In particular, the last vertex incident to the red edge is extremal. ◀

► **Observation 14.** *If G_n, \dots, G_1 are the trigraphs of a 1-contraction sequence of G , their underlying graphs form a chain for the induced subgraph relation, i.e. G_{i-1} is an induced subgraph of G_i for $i \in [2, n]$.*

Proof. We describe how to construct the induced subgraph of G by picking a representative in G for each vertex in G_i .

Consider a vertex v of G_i , either it has only black incident edges and we may pick any vertex of G that was contracted into v , or it has exactly one incident red edge vw , in which case we can pick the endpoints of some edge of G that is between the vertices contracted into v and the vertices contracted into w .

This construction picks only one vertex of G per vertex of G_i because the red degree is at most one meaning we do not pick more than once. ◀

This property only holds for 1-contraction sequences (e.g. contracting non-adjacent vertices of a matching creates a path on 3 vertices which is not an induced subgraph). It allows to view the contraction sequence as a sequence of vertex deletions. In particular, underlying graphs of the trigraphs of a 1-contraction sequence are also permutation graphs and one of their permutation diagrams is the induced permutation diagram obtained by seeing G_i as an induced subgraph of G . Observe that this diagram preserves the relative order of vertices that are not deleted. This will be important for our inductive reasoning since we can keep the same diagram while decomposing the graph. We denote by $\sigma[G_i]$ the ordering induced by G_i seen as an induced subgraph of G .

► **Corollary 15.** *Let G be a graph and G_n, \dots, G_1 a 1-contraction sequence of G . The realiser constructed from G_n, \dots, G_1 using Lemma 11 has the property that, for every $i > 1$, the vertices being contracted from G_i to G_{i-1} are consecutive for $(\sigma[G_i], \tau[G_i])$, and endpoints of red edges are consecutive for $(\sigma[G_i], \tau[G_i])$.*

Conversely, for any realiser (σ', τ') , there exists a 1-contraction sequence G_n, \dots, G_1 such that for every $i > 1$ the contracted vertices of G_i are consecutive in $(\sigma'[G_i], \tau'[G_i])$ and for any $xy \in R(G_i)$, x and y are consecutive in $(\sigma'[G_i], \tau'[G_i])$.

Proof. Suppose we contract $a, b \in V(G_i)$ into vertex $c \in V(G_{i-1})$. Let C be the set of vertices of G contracted into c , and A, B those contracted into a, b . By Lemma 11, A, B and C are all intervals of (σ, τ) . Since $C = A \cup B$, we may assume without loss of generality they

are all intervals of σ . Hence, a and b are consecutive for $\sigma[G_i]$. Similarly, the set of vertices of G contracted into a fixed red edge is an interval of (σ, τ) . Since the sets of vertices of G incident to the red edge are also intervals, this means the endpoints of the red edge in G_i are consecutive.

Now, consider any realiser (σ', τ') for graph G , and let us show the second part of the lemma by induction on the modular decomposition of G . The result holds trivially for leaves of the decomposition. Let us then consider an internal node q of the decomposition of G and consider the subgraphs H_1, \dots, H_k of G induced by the children of q . Now, since each H_i is a strong module of G , they each form a common interval of (σ', τ') by Observation 3. They also have twin-width 1, so our induction hypothesis along with Lemma 5 yields that we can contract each child fully by respecting the consecutivity properties on $(\sigma'[H_i], \tau'[H_i])$, and thus on (σ', τ') . Then, if q is a degenerate node, we can always find a pair of consecutive twins and contract them. Otherwise, q is prime, and as such it admits a unique realiser (up to symmetry), which is given by Lemma 11, for which the contraction sequence satisfies the desired properties by the first part of the lemma. ◀

4 Linear-time recognition algorithm

A realiser (σ, τ) , if it exists, can be found in linear time [24]. From the Corollaries 6, 12 and 15, we deduce that the algorithm of [10] can be implemented in time $O(n^2 + m)$. Indeed, for every prime node of the modular decomposition, it suffices to compute σ and τ for its quotient graph. At this point, we have restricted the set of possible first contractions to the set all pairs of consecutive vertices, and we may then simulate contraction sequences efficiently using the fact that new vertices to be contracted can be found in constant time, since they are consecutive. In order to obtain a linear time algorithm, we will instead guess the end of the sequence, and try to extend the sequence from its end, see Figure 2.

► **Lemma 16.** *Let G be a prime permutation graph of twin-width 1, which admits a 1-contraction sequence where an extremal vertex s is last to become incident to a red edge. Let also $G^1 = G[N(s)]$ and $G^2 = G[\overline{N}(s)]$.*

Then, there is a module M of G^i for some i that has a unique splitter $s' \in V(G^{3-i})$, such that $G[M \cup \{s'\}]$ admits a 1-contraction sequence where s' is the last vertex to be incident to the red edge, where:

- (i) *either M is a pair of consecutive twins in G^i and there are no prime nodes in the modular decompositions of G^1 and G^2 ,*
- (ii) *or M corresponds to the subtree rooted at a prime node p in the modular decomposition, and p is the unique prime node in the modular decompositions of G^1 and G^2 .*

Moreover, let $k = |V(G) - (M \cup \{s'\})|$, there is an ordering $\pi : [k] \rightarrow V(G) - (M \cup \{s'\})$ such that for all $i \in [k]$, $M \cup \{s'\} \cup \pi([i])$ forms an interval of σ and two intervals of τ for a realiser (σ, τ) where σ is the linear ordering for which s is extremal.

Proof. Let (σ, τ) be a realiser of G and s be an extremal vertex for σ which is the last vertex incident to the red edge in some 1-contraction sequence of G . Such a vertex exists due to Corollary 13. Let $G^1 = G[N(s)]$ and $G^2 = G[\overline{N}(s)]$ and observe that since G is prime, G^1 and G^2 are not empty. As s is extremal for σ , adjacent to $V(G^1)$ and non-adjacent to $V(G^2)$, $V(G^1)$ and $V(G^2)$ must be disjoint intervals of τ .

▷ **Claim 17.** In any contraction sequence as considered, we only contract pairs of vertices of G^1 or pairs of vertices of G^2 until we reach the 3-vertex trigraph.

Proof. Due to the assumption on s being the last vertex incident to a red edge, we cannot contract a vertex of G^1 with a vertex of G^2 , unless G^1 and G^2 both have been contracted to a single vertex, because they are split by s . \triangleleft

▷ **Claim 18.** For any non-trivial module M of G^i there is a splitter $s' \in V(G^{3-i})$. Furthermore, for any splitter $s' \in V(G^{3-i})$ of M there are a, b in M such that $\sigma(a) < \sigma(s') < \sigma(b)$.

Proof. Indeed, M cannot be a module of G , as G is a prime graph, so it must have a splitter s' in $G - V(G^i)$. Also note that s is not a splitter of M by definition of G^i . Furthermore, since s' is a splitter it must be adjacent to some $a \in M$ and non-adjacent to some $b \in M$. Since $V(G^1)$ and $V(G^2)$ are intervals of τ , we get that $\sigma(a) < \sigma(s') < \sigma(b)$ or $\sigma(a) < \sigma(s') < \sigma(b)$ by definition of the permutation diagram. \triangleleft

The following observation will allow to simplify the analysis.

► **Observation 19.** If X induces a prime subgraph of H , and $t \in V(H) \setminus X$ is a splitter of X , then we can both extract a pair of vertices of X that are adjacent and split by t , and a pair of vertices of X that are non-adjacent and split by t .

Proof. Consider the cut $C = (X \cap N(t), X - N(t))$ in $H[X]$, since $H[X]$ is prime both $H[X]$ and $\overline{H[X]}$ are connected, yielding both an edge and a non-edge across C . \blacktriangleleft

▷ **Claim 20.** There cannot be disjoint non-trivial strong modules in G^i .

Proof. We proceed by contradiction and consider M, M' two maximal disjoint non-trivial strong modules of G^i , without loss of generality $i = 1$. We first show that, in G , M and M' must have distinct splitters from $V(G^2)$, and then conclude that this contradicts the assumption that G has a 1-contraction sequence respecting Claim 17.

Using the fact that $V(G^1)$ and $V(G^2)$ are intervals of τ and the common interval characterisation of strong modules of G^1 given by Observation 3, we deduce that a splitter t , which necessarily belongs to $V(G^2)$, cannot split both M and M' . Indeed, we have $\sigma(M) < \sigma(M')$ or $\sigma(M') < \sigma(M)$, so if t is a splitter of M there exist $a, b \in M$ such that $\sigma(a) < \sigma(t) < \sigma(b)$ as seen in the previous claim. In particular, t is not a splitter of M' .

Now because M and M' are non-trivial modules of G^1 , they each have a splitter in G^2 . Let t and t' be such splitters. By maximality, M and M' correspond to subtrees rooted at siblings of the same node n in the modular decomposition of G^1 . Let us first consider the case when n is a prime node. In this case, Lemma 7 guarantees a red edge for $Q(n)$, and the two splitters t, t' forbid the existence of a contraction sequence with consecutivity properties as guaranteed by Lemma 11 and Corollary 15.

We may now assume that n is a degenerate node, and exhibit an induced subgraph H of G that does not admit a 1-contraction sequence respecting our assumptions. This will contradict the existence of such a sequence for the whole G , as its restriction to H would also be valid (as in Observation 4). The graph H consists of splitters t, t' , as well as two pairs of twins $a, b \in M$ and $a', b' \in M'$ distinguished by t, t' respectively. Then, vertices a, b, a', b' induce a cograph whose corresponding cotree is a complete binary tree of depth 2. In other words, (a, b) and (a', b') are present if and only if $\{a, b\}$ is adjacent to $\{a', b'\}$.

We first reduce M and M' so as to induce exactly the quotient graph of their corresponding node in the modular decomposition (recall that this is well defined because they are strong) by picking one vertex in each subtree below this node. We may ensure that t and t' still split this subset of vertices (e.g. by picking extremal vertices of the interval of σ corresponding to M and M' , recalling Observation 3).

Now we observe that if M (resp. M') has a degenerate root node, it is of the opposite type of the node n , and any pair of vertices of M (resp. M') that is split by t (resp. t') can be taken for our subgraph H . Otherwise, M (resp. M') has a prime root node, and we apply Observation 19 to obtain a pair of vertices that is split and has the required adjacency relation.

Let us now show that H does not admit a 1-contraction sequence satisfying our assumptions. Indeed, t and t' may only be contracted together, because they are the only vertices in $V(G^2)$ of our subgraph (Claim 17), but they are split by at least two vertices in $V(G^1)$. Now, any order of contractions on a, b, a', b' will create two red edges. More precisely, contracting a vertex from each nontrivial module creates two red edges, unless one module was already reduced to a single vertex, in which case it has a red edge to its splitter from $V(G^2)$. This will inevitably contradict Lemma 7. \triangleleft

\triangleright Claim 21. If G^i has a prime node p in its modular decomposition, the first red edge must appear in the module M corresponding to the subtree rooted at p . In this case, M has exactly one splitter s' from G^{3-i} and s' should not be incident to a red edge until M is contracted to a single vertex.

Proof. Let p be a prime node of the modular decomposition of G^i and M the module corresponding to the subtree rooted at p . Without loss of generality assume $i = 1$. We first make the observation that, since there cannot be two disjoint nontrivial strong modules in G^1 by Claim 20, and since prime graphs have at least 4 vertices, there are vertices introduced by node p .

We first consider the case when a red edge incident to a vertex of G^2 is created before we fully contract M . This red edge has to remain incident to a vertex of G^2 due to Claim 17. Contracting a vertex introduced in prime node p will create another red edge with both endpoints in G^1 , which is thus different and contradicts Lemma 7.

We now deal with the case where no red edge incident to a vertex of G^2 appears before we fully contract M . Assume the first red edge does not have both endpoints in M . Observe that the first contraction cannot be between vertices that are both not in M but have a different adjacency to it because M is a module on at least 4 vertices, and this would create a red edge to all of them. Observe also that if the first contraction had involved a vertex of M , there would also be a red edge with both endpoints in M . Indeed, on the one hand, if both contracted vertices were in M , red edges with both endpoints in G^i would have both endpoints in M because it is a module. On the other hand, M is also a strong module with a prime node at its root, so all of its vertices have a mixed adjacency to the rest of M , contrary to vertices outside M . We conclude that the first contraction is between two vertices outside of M , so the corresponding red edge has both endpoints outside of M . This has the following implications: both contracted vertices have an homogeneous adjacency to M , but they also have a unique splitter in G^1 , this implies we cannot contract the red edge before contracting the subgraph corresponding to p . Indeed, the two vertices incident to the red edge must have opposite adjacency relations with respect to M . Since contracting M will force the creation of a second red edge, this contradicts Lemma 7.

M is a non-trivial strong module of G^1 so it has at least one splitter in G^2 . From Lemma 7 and Claim 17, it follows that a red edge incident to this splitter cannot appear before M is contracted to a single vertex. Finally, only one red edge may appear when M is contracted to a single vertex meaning the splitter must be unique. \triangleleft

In particular, this shows that only one of G^1 and G^2 can have a prime node in their modular decomposition. We can moreover deduce that if there is a prime node in G^i , it must be unique. Indeed, a red edge has to appear in some M corresponding to the subtree of the

prime node that is deepest in the modular decomposition of G^i . At any step of the sequence, M contains a red edge, until it is a single vertex with a red edge towards its splitter. Then, vertices introduced in any higher prime node cannot be contracted on such red edges without producing an additional red edge, which contradicts Lemma 7.

We are now ready to define M . Consider the first red edge appearing between $X_1 \subseteq V(G^1)$ and $X_2 \subseteq V(G^2)$ in a 1-contraction sequence as considered, and assume without loss of generality that it stems from a contraction in G^1 . Then, X_2 can only consist in a single vertex s' due to Claim 17, Lemma 7, and the fact that this is the first red edge with an endpoint in both G^1 and G^2 . Then, X_1 is included in a module of G^1 which admits as unique splitter s' . In the case when there is no prime node, X_1 induces a cograph, so we may then take M to be any pair of consecutive twins $a, b \in X_1$, which are also consecutive twins in G^1 by Corollary 15, distinguished by s' (because G is prime and s' is the only splitter of X_1).

Otherwise, we can take M and s' as guaranteed by Claim 21 on G^1 and G^2 . The restriction of our contraction sequence to $G[M \cup s']$ yields a 1-contraction sequence in which s' becomes incident to a red edge last, as desired.

Having defined M , we are ready to show the following.

▷ **Claim 22.** There exists a 1-contraction sequence of G that starts by contracting only vertices of M until it is reduced to a single vertex. In particular, this creates a red edge towards s' at the last contraction.

Proof. We can extend the contraction sequence that contracts M first into a sequence contracting the smallest interval I of $\sigma[V(G^1)]$ containing X_1 because X_1 is not split by any splitter from G^2 other than s' , and it does not have disjoint non trivial strong modules. After contraction of M to a single vertex, the remaining vertices of I induce a cograph that is not split by any vertex of G^2 other than s' . We can then proceed with the rest of the contraction of G simply by contracting remaining vertices according to our initial contraction sequence. ◁

At this point, in both cases, we know M forms a common interval for $(\sigma[V(G^1)], \tau[V(G^1)])$, and that there is some 1-sequence for G that first contracts M and in which s is the last vertex to be incident to a red edge. We deduce that $M \cup \{s'\}$ satisfies the setting of Corollary 15, and any further contractions must be consecutive to M due to Corollary 15, ensuring the existence of π . ◀

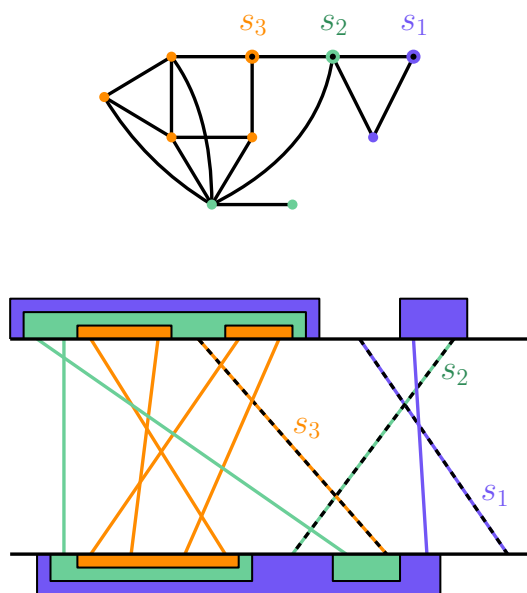
The crucial observation now is that π can easily be computed greedily using the realiser.

► **Lemma 23.** *Given a permutation diagram of a prime permutation graph G and an extremal vertex s of G , we can produce in time linear in the number of vertices of G a 1-contraction sequence such that s is the last vertex of G to be incident to the red edge, or conclude that no such sequence exists.*

Proof. This is essentially an implementation of the result of Lemma 16, we reuse similar notations and provide an illustration in Figure 2. The algorithm works as follows.

We initialise indices at the endpoints of the intervals A and B of τ that are split by s , and at the endpoints of the interval $C = V(G) - \{s\}$ of σ . We add s to the global list Π and mark it as a splitter.

While there is one endpoint v of C that is also an endpoint of A or an endpoint of B , we add v to the global list Π and update indices describing A, B , and C in accordance to the removal of v . We call such a vertex v *doubly extremal*.



■ **Figure 2** An example of a twin-width 1 graph and its permutation diagram. Shown here is the decomposition of Lemma 23 starting from extremal vertex s_1 . The segments of splitters s_1, s_2, s_3 are represented in dashed blue, green and orange respectively. Then, intervals of the corresponding colour are the intervals A, B, C produced during the iteration in which the splitter is considered. A and B induce subgraphs G^i from Lemma 16. Then, segments of the same colour as a splitter correspond to doubly extremal vertices eliminated in the corresponding step.

If the intervals have become empty, return Π , otherwise, if either A or B contains a single vertex, we set s' to be this vertex and M to be the other interval, and apply the algorithm recursively to $G[M \cup \{s'\}]$ with s' as the last vertex incident to a red edge. Finally, if both A and B contain more than one vertex, we reject.

We can use marked vertices to deduce a contraction sequence: we iterate through Π , starting from the last vertices added. When reaching a marked vertex, we contract the red edge. Otherwise, we contract the current vertex to the vertex of the red edge with the same adjacency to the next marked vertex.

If the algorithm does not reject, let us show by induction on the recursive calls that the list Π , along with its marked vertices, describes a valid order of contractions for G . First, we clarify the starting red edge that we consider in the part of the contraction sequence built at this step of the recursion. In the case when we emptied A and B (so M and s' are not defined), take the last vertex in A and the last vertex in B as a starting red edge. We move to the case where M and s' are defined. Here, by induction hypothesis, the algorithm finds a 1-contraction sequence with s' as the last vertex incident to a red edge for $G[M \cup \{s'\}]$. This contraction sequence for $G[M \cup \{s'\}]$ gives a partial 1-contraction sequence for G (see Observation 4) yielding a red edge between M and s' , with all other vertices of G not contracted yet.

We now check that each vertex added to Π at this step can be contracted in the order described by Π . This is the case because the vertex does not split the vertices of A contracted before it, nor the vertices of B contracted before it. By contracting it to the vertex of the red edge with the same adjacency to s , we create no additional red edge. In particular, s is still not incident to a red edge. Once we contracted all vertices of A and B , the red edge can be safely contracted. Indeed, s is the unique splitter of $A \cup B$.

If the algorithm rejects, it contradicts the existence of π guaranteed by Lemma 16. Indeed, if for all i , the $M \cup \{s'\} \cup \pi([i])$ are intervals as claimed, it must be that for each i , $\pi(i)$ is extremal on $\sigma[M \cup \{s'\} \cup \pi([i])]$ and on $\tau[(M \cup \{s'\} \cup \pi([i])) \cap V(G^j)]$ for some $j \in \{1, 2\}$. In particular, all vertices that are not in $M \cup \{s'\}$ will become doubly extremal if π exists. Note that this is independent of the order in which they are contracted. Indeed, a vertex remains (doubly) extremal while we delete other vertices.

We conclude that G has twin-width more than 1. ◀

► **Theorem 24.** *We can decide in linear time if a given graph G has twin-width at most 1.*

Proof. We first compute a modular decomposition of G and realisers for the quotient graphs of prime nodes [24]¹, or conclude that G is not a permutation graph and therefore not a graph of twin-width 1. It is well-known that the total size of the quotient graphs of prime nodes is linear in the size of the original graph.

We then check all prime nodes with corresponding quotient graph H as follows:

Guess an extremal vertex s (out of at most 4) that can be the last vertex incident to the red edge in a 1-contraction sequence of H , and then apply the recursive algorithm of Lemma 23.

If all prime nodes have twin-width 1, we obtained a contraction sequence for all of them and they can be combined to form a contraction sequence of G using Corollary 6.

All of the above procedures can be implemented to run in linear time. ◀

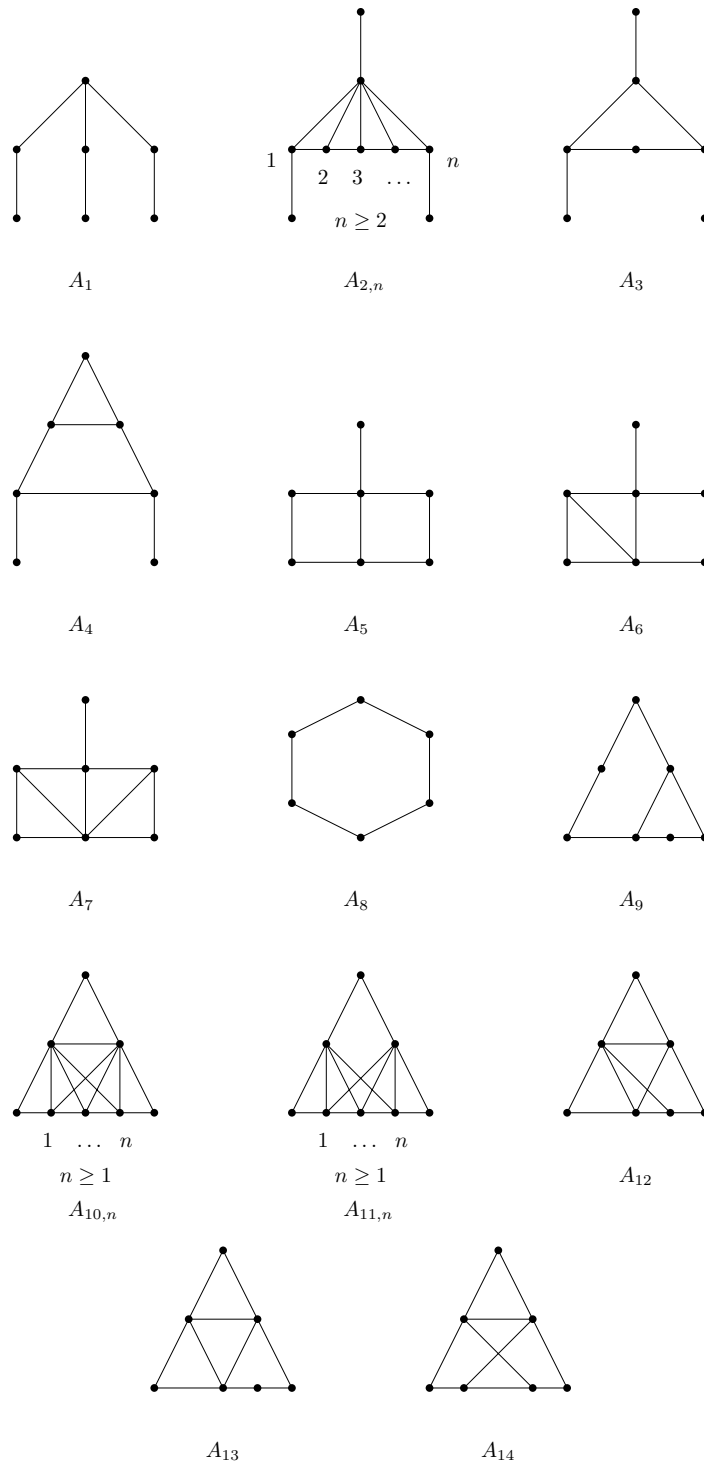
References

- 1 Jungho Ahn, Debsoumya Chakraborti, Kevin Hendrey, and Sang-il Oum. Twin-width of subdivisions of multigraphs, 2024. doi:10.48550/arXiv.2306.05334.
- 2 Jungho Ahn, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. Bounds for the twin-width of graphs. *SIAM J. Discret. Math.*, 36(3):2352–2366, 2022. doi:10.1137/21m1452834.
- 3 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 Lokshantov, 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.
- 4 Jakub Balabán, Robert Ganian, and Mathis Rocton. Twin-width meets feedback edges and vertex integrity. *CoRR*, abs/2407.15514, 2024. doi:10.48550/arXiv.2407.15514.
- 5 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.
- 6 Anne Bergeron, Cédric Chauve, Fabien de Montgolfier, and Mathieu Raffinot. Computing common intervals of K permutations, with applications to modular decomposition of graphs. *SIAM J. Discret. Math.*, 22(3):1022–1039, 2008. doi:10.1137/060651331.
- 7 Benjamin Bergognoux, Jakub Gajarský, Grzegorz Guspiel, Petr Hliněný, Filip Pokrývka, and Marek Sokolowski. Sparse graphs of twin-width 2 have bounded tree-width. In Satoru Iwata and Naonori Kakimura, editors, *34th International Symposium on Algorithms and Computation, ISAAC 2023, December 3-6, 2023, Kyoto, Japan*, volume 283 of *LIPICs*, pages 11:1–11:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.ISAAC.2023.11.

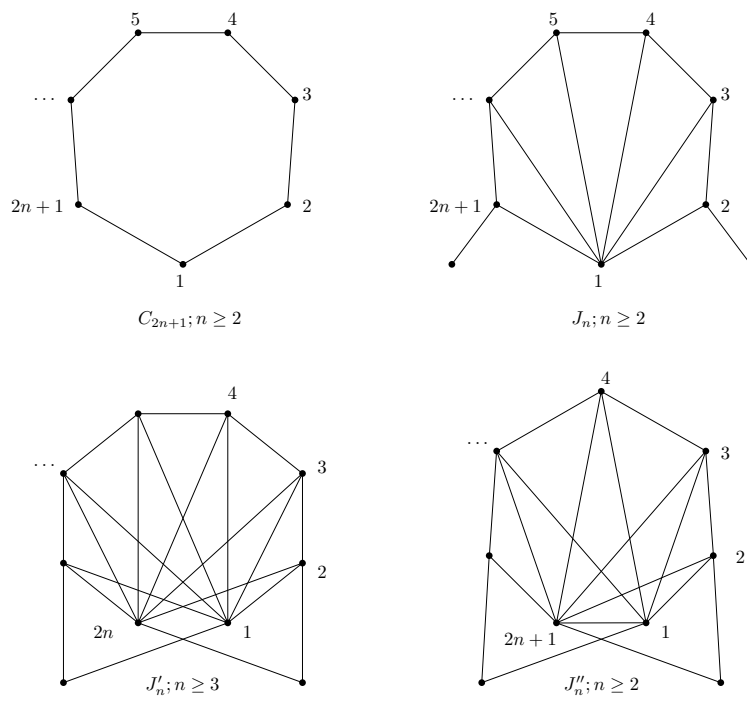
¹ In fact, it would be more reasonable to compute a permutation diagram and deduce a modular decomposition via common intervals for a practical implementation.

- 8 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Torunczyk. Twin-width IV: ordered graphs and matrices. *J. ACM*, 71(3):21, 2024. doi:10.1145/3651151.
- 9 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.
- 10 Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, Stéphan Thomassé, and Rémi Watrigant. Twin-width and polynomial kernels. *Algorithmica*, 84(11):3300–3337, 2022. doi:10.1007/s00453-022-00965-5.
- 11 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: Tractable FO model checking. *J. ACM*, 69(1):Art. 3, 46, 2022. doi:10.1145/3486655.
- 12 Christian Capelle, Michel Habib, and Fabien de Montgolfier. Graph decompositions and-factorizing permutations. *Discret. Math. Theor. Comput. Sci.*, 5(1):55–70, 2002. doi:10.46298/DMTCS.298.
- 13 Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164(1):51–229, July 2006. doi:10.4007/annals.2006.164.51.
- 14 Derek G. Corneil, Michel Habib, Jean-Marc Lanlignel, Bruce A. Reed, and Udi Rotics. Polynomial-time recognition of clique-width ≤ 3 graphs. *Discret. Appl. Math.*, 160(6):834–865, 2012. doi:10.1016/J.DAM.2011.03.020.
- 15 Derek G Corneil, Stephan Olariu, and Lorna Stewart. Asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 10(3):399–430, 1997. doi:10.1137/S0895480193250125.
- 16 Fabien de Montgolfier. *Décomposition modulaire des graphes. Théorie, extension et algorithmes. (Graph Modular Decomposition. Theory, extension and algorithms)*. PhD thesis, Montpellier 2 University, France, 2003. URL: <https://tel.archives-ouvertes.fr/tel-03711558>.
- 17 Ben Dushnik and E. W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63(3):600–610, July 1941.
- 18 Tibor Gallai. Transitiv orientierbare graphen. *Acta Mathematica Academiae Scientiarum Hungaricae*, 18:25–66, 1967. German.
- 19 Colin Geniet and Stéphan Thomassé. First order logic and twin-width in tournaments. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands*, volume 274 of *LIPICs*, pages 53:1–53:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.ESA.2023.53.
- 20 M.C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, 1980.
- 21 Sylvain Guillemot and Dániel Marx. Finding small patterns in permutations in linear time. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 82–101. SIAM, 2014. doi:10.1137/1.9781611973402.7.
- 22 Michel Habib and Christophe Paul. A survey of the algorithmic aspects of modular decomposition. *Comput. Sci. Rev.*, 4(1):41–59, 2010. doi:10.1016/J.COSREV.2010.01.001.
- 23 Ekkehard Köhler. *Graphs Without Asteroidal Triples*. PhD thesis, Technischen Universität Berlin, 1999.
- 24 Ross M. McConnell and Jeremy P. Spinrad. Modular decomposition and transitive orientation. *Discret. Math.*, 201(1-3):189–241, 1999. doi:10.1016/S0012-365X(98)00319-7.
- 25 Szymon Torunczyk. Flip-width: Cops and robber on dense graphs. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 663–700. IEEE, 2023. doi:10.1109/FOCS57990.2023.00045.

A Appendix



■ **Figure 3** Minimal graphs containing an asteroid triple.



■ **Figure 4** Complements of minimal graphs containing a $(2k + 1)$ -asteroid for $k > 1$.