NodeTrix Planarity Testing with Small Clusters

Emilio Di Giacomo\textsuperscript{1}, Giuseppe Liotta\textsuperscript{1},
Maurizio Patrignani\textsuperscript{2}, Alessandra Tappini\textsuperscript{1}

\textsuperscript{1} Università degli Studi di Perugia, Italy
\{emilio.digiacomo,giuseppe.liotta\}@unipg.it
alessandra.tappini@studenti.unipg.it
\textsuperscript{2} Roma Tre University, Italy
patrigna@dia.uniroma3.it

Abstract. We study the NodeTrix planarity testing problem for flat clustered graphs when the maximum size of each cluster is bounded by a constant $k$. We consider both the case when the sides of the matrices to which the edges are incident are fixed and the case when they can be arbitrarily chosen. We show that NodeTrix planarity testing with fixed sides can be solved in $O(k^3 + 3^2 n^3)$ time for every flat clustered graph that can be reduced to a partial 2-tree by collapsing its clusters into single vertices. In the general case, NodeTrix planarity testing with fixed sides can be solved in $O(n^3)$ time for $k = 2$, but it is NP-complete for any $k \geq 3$. NodeTrix planarity testing remains NP-complete also in the free side model when $k > 4$.

1 Introduction

Motivated by the need of visually exploring non-planar graphs, hybrid planarity is one of the emerging topics in graph drawing (see, e.g., [1,2,3,8]). A hybrid planar drawing of a non-planar graph suitably represents in restricted geometric regions those dense subgraphs for which a classical node-link representation paradigm would not be visually effective. These regions are connected by edges that do not cross each other. Different representation paradigms for the dense subgraphs give rise to different types of hybrid planar drawings.

Angelini et al. [1] consider hybrid planar drawings where dense portions of the graph are represented as intersection graphs of sets of rectangles and study the complexity of testing whether a non-planar graph admits such a representation. In the context of social network analysis, Henry et al. [8] introduce NodeTrix representations, where the dense subgraphs are represented as adjacency matrices. Batagelj et al. [2] study the question of minimizing the size of the matrices in a NodeTrix representation of a graph while guaranteeing the planarity of the edges that connect different matrices. While Batagelj et al. can choose the subgraphs to be represented as matrices, Da Lozzo et al. [3] consider the problem of testing whether a flat clustered graph (i.e. a graph with clusters and no sub-clusters) admits a NodeTrix planar representation. In the paper of Da Lozzo et al. each cluster must be represented by a different adjacency matrix and the
inter-cluster edges are represented as non-intersecting simple Jordan arcs. They prove that NodeTrix planarity testing for flat clustered graphs is NP-hard even in the constrained case where for each matrix it is specified which inter-cluster edges must be incident on the top, or on the left, or on the bottom, or on the right side.

Motivated by these hardness results, in this paper we study whether NodeTrix planarity testing can be efficiently solved when the size of the clusters is not “too big”. More precisely, we consider flat clustered graphs whose clusters have size bounded by a fixed parameter $k$ and we want to understand whether the NodeTrix planarity testing problem is fixed parameter tractable, i.e. it can be solved in time $f(k)T(n)$, where $T(n)$ is a polynomial in $n$ and $f(k)$ is a function that depends only on $k$. Our main results can be listed as follows:

- We describe an $O(k^{3k^3/2}n^3)$-time algorithm to test NodeTrix planarity with fixed sides for flat clustered graphs that are partial 2-trees. Informally, a flat clustered graph $G$ is a partial 2-tree if the graph obtained by collapsing every cluster of $G$ into a single vertex is a partial 2-tree.
- When the flat clustered graph is not a partial 2-tree, NodeTrix planarity testing with fixed sides can still be solved in $O(n^3)$ time for $k = 2$, but it becomes NP-complete for any larger value of $k$.
- Finally, we extend the above hardness result to the free sides model and show that NodeTrix planarity testing remains NP-complete when the maximum cluster dimension is larger than four. This is done by proving that NAE3SAT is NP-complete even for triconnected Boolean formulas, which may be a result of independent interest.

Our polynomial-time solution solves a special type of the planarity testing problem where the order of the edges around each vertex is suitably constrained to take into account the fact that a vertex of a matrix $M$ has four copies along the four sides of $M$. It may be worth recalling that Gutwenger et al. [6] have considered an apparently similar problem. Namely, they studied planarity testing where the order of the edges around the vertices may not be arbitrarily permuted. Unfortunately, not only our problem does not fall in any of the cases addressed by Gutwenger et al., but it does not seem solvable by introducing a gadget of polynomial size that models the embedding constraints at each vertex. This characteristic associates NodeTrix planarity testing with other known variants of planarity testing, including clustered planarity, where the use of gadgets of polynomial size has been so far an elusive goal.

The rest of the paper is organized as follows. Preliminary definitions are in Section 2. Sections 3 and 4 describe a polynomial time algorithm for clustered 2-trees with bounded cluster-size. In Section 5 we show that for general flat clustered graphs and fixed sides NodeTrix planarity testing can be solved in polynomial time for $k = 2$ but it is NP-complete for $k \geq 3$. In the same section we extend this completeness result to NodeTrix planarity testing of flat clustered graphs with free sides. Finally open problems can be found in Section 6. Some proofs can be found in the appendix.
2 Preliminaries

We assume familiarity with basic definitions of graph theory and graph drawing and in particular with the notions of block-cut-vertex tree and of SPQR-tree (see, e.g., [17]).

A flat clustered graph \( G = (V, E, C) \) is a simple graph with vertex set \( V \), edge set \( E \), and a partition \( C \) of \( V \) into sets \( V_1, \ldots, V_k \), called clusters. An edge \((u,v) \in E \) with \( u \in V_i \) and \( v \in V_j \) is an intra-cluster edge if \( i = j \) and it is an inter-cluster edge if \( i \neq j \).

A NodeTrix representation of a flat clustered graph \( G \) is such that: (i) Each cluster \( V_i \) with \(|V_i| = 1 \) (called trivial cluster) is represented as a distinct point in the plane. (ii) Each cluster \( V_i \) with \(|V_i| > 1 \) (called non-trivial cluster) is represented by a symmetric adjacency matrix \( M_i \) (with \(|V_i| \) rows and columns), where \( M_i \) is drawn in the plane so that its boundary is a square with sides parallel to the coordinate axes. (iii) There is no intersection between two distinct matrices or between a point representing a vertex and a matrix. (iv) Each intra-cluster edge of a cluster \( V_i \) is represented by the adjacency matrix \( M_i \). (v) Each inter-cluster edge \((u,v) \) with \( u \in V_i \) and \( v \in V_j \) is represented by a simple Jordan arc connecting a point on the boundary of matrix \( M_i \) with a point on the boundary of matrix \( M_j \), where the point on \( M_i \) (resp. of \( M_j \)) associated with \( u \) (resp. with \( v \)).

A NodeTrix representation of a flat clustered graph \( G \) is planar if there is no intersection between any two inter-cluster edges (except possibly at common end-points) nor an intersection between an inter-cluster edge and a matrix. A flat clustered graph is NodeTrix planar if it admits a planar NodeTrix representation. Fig. 1(a) is an example of a NodeTrix planar representation.

A formal definition of the problem investigated in the paper is as follows. Let \( G = (V, E, C) \) be a flat clustered graph with \( n \) vertices and let \( k \) be the maximum cardinality of a cluster in \( C \). Clustered graph \( G \) is NodeTrix planar with fixed sides if it has a NodeTrix planar representation where for each inter-cluster edge, the sides of matrices it attaches to is specified as part of the input; \( G \) is NodeTrix planar with free sides if the sides of the matrices to which inter-cluster edges attach can be arbitrarily chosen.

Let \( M_i \) be the matrix representing cluster \( V_i \) in a NodeTrix representation of \( G \); let \( v \) be a vertex of \( V_i \) and let \((u,v) \) be an inter-cluster edge. Edge \((u,v) \) can intersect the boundary of \( M_i \) in four points \( p_{v,T}, p_{v,B}, p_{v,L}, \) and \( p_{v,R} \) since the row and column that represent \( v \) in \( M_i \) intersect the four sides of the boundary of \( M_i \). We call such points the top copy, bottom copy, left copy, and right copy of \( v \) in \( M_i \), respectively.

A side assignment for \( V_i \in C \) specifies for each inter-cluster edge whether the edge must attach to the matrix \( M_i \) representing \( V_i \) in its top, left, right, or bottom side. More precisely, a side assignment is a mapping \( \phi_i : \bigcup_{j \neq i} E_{i,j} \rightarrow \{T, B, L, R\} \), where \( E_{i,j} \) is the set of inter-cluster edges between the clusters \( V_i \) and \( V_j \) (\( V_i \) and \( V_j \) are adjacent if \( E_{i,j} \neq \emptyset \)). A side assignment for \( C \) is a set \( \Phi \) of side assignments for each \( V_i \in C \).
We denote as \( G = (V,E,C,\Phi) \) a flat clustered graph \( G = (V,E,C) \) with a given side assignment \( \Phi = \{\phi_1,\phi_2,\ldots,\phi_{|C|}\} \). Let \( \Gamma \) be a NodeTrix representation of \( G \) such that, for every inter-cluster edge \( e = (u,v) \in E \) with \( u \in V_i \) and \( v \in V_j \), the incidence points of \( e \) with the matrices \( M_i \) and \( M_j \) representing \( V_i \) and \( V_j \) in \( \Gamma \) are exactly the points \( p_{u,\phi_i(e)} \) and \( p_{v,\phi_j(e)} \), respectively. We call \( \Gamma \) a NodeTrix representation of \( G \) consistent with \( \Phi \). We say that \( G = (V,E,C,\Phi) \) is NodeTrix planar if it admits a NodeTrix planar representation consistent with \( \Phi \).

An inter-cluster edge is heavy if both its end-vertices belong to non-trivial clusters. It is light otherwise. A flat clustered graph is light if every inter-cluster edge is light. A 1-subdivision of a heavy edge \( e = (u,v) \) of a flat clustered graph \( G = (V,E,C) \) replaces \( e \) with a path \( u_0 = u, u_1, u_2 = v \) and defines a new flat clustered graph \( G' = (V',E',C') \), where \( V' = V \cup \{u_1\} \), \( E' = E/e \cup \{(u_0,u_1),(u_1,u_2)\} \), and \( C' = C \cup \{u_1\} \). The light reduction of \( G \) is the flat clustered graph \( G'' \) obtained by performing a 1-subdivision of every heavy inter-cluster edge of \( G \). A consequence of Theorem 1 in [5] about the edge density of NodeTrix planar graphs, is that the light reduction \( G' \) of a NodeTrix planar flat clustered graph \( G \) has \( O(|V|) \) vertices and \( O(|V|) \) inter-cluster edges.

**Property 1.** A flat clustered graph \( G \) is NodeTrix planar if and only if its light reduction \( G' \) is NodeTrix planar.

Based on Property 1 in the remainder we shall assume that flat clustered graphs are always light and we call them clustered graphs, for short.

The frame of a clustered graph \( G = (V,E,C) \) is the graph \( F \) obtained by collapsing each cluster \( V_i \in C \) with \( |V_i| > 1 \), into a single vertex \( c_i \) of \( F \), called the representative vertex of \( V_i \) in \( F \). Let \( c_i \) and \( c_j \) be the two representative vertices of \( V_i \) and \( V_j \) in \( F \), respectively. For every inter-cluster edge connecting a vertex of \( V_i \) to a vertex of \( V_j \) in \( G \) there is an edge in \( F \) connecting \( c_i \) and \( c_j \). Observe that the frame graph \( F \) of \( G \) is in general a multigraph; however, \( F \) is simple when \( G \) is light.

Since the NodeTrix planarity of a clustered graph implies the planarity of its frame graph, we will test NodeTrix planarity only on those clustered graphs that have a planar frame.

A 2-tree is a graph recursively defined as follows: (i) an edge is a 2-tree; (ii) the graph obtained by adding a vertex \( v \) to a 2-tree \( G \) and by connecting \( v \) to two adjacent vertices of \( G \) is a 2-tree. A (planar) graph is a partial 2-tree if it is a subgraph of a (planar) 2-tree. A biconnected partial 2-tree is a series-parallel graph. A clustered graph is a partial 2-tree if its frame is a partial 2-tree. We will sometimes talk about series-parallel clustered graphs when their frames are series parallel.

### 3 NodeTrix Representations and Wheel Reductions

The polynomial-time algorithms described in Sections 4 and 5 are based on decomposing the planar frame \( F \) of a clustered graph \( G = (V,E,C,\Phi) \) into its biconnected components and storing them into a block-cut-vertex tree. We
process each block of $F$ by using an SPQR decomposition tree that is rooted at a reference edge and visited from the leaves to the root. For each visited node $\mu$ of the decomposition tree of a block of $F$ we test whether the subgraph of $G$ whose frame is the pertinent graph of $\mu$ satisfies the planar constraints imposed by the side assignment on the inter-cluster edges. A key ingredient to efficiently perform the test at $\mu$ is the notion of wheel replacement.

Let $G = (V,E,C,\Phi)$ be a clustered graph with side assignment $\Phi$ and let $V_i \in C$ be a cluster with $k > 1$ vertices. $V_i$ admits $k!$ permutations of its vertices and we associate a suitable graph to each such permutation. Let $\pi_i = v_0, v_1, \ldots, v_{k-1}$ be a permutation of the vertices of $V_i$. The wheel of $V_i$ consistent with $\pi_i$ is the wheel graph consisting of a vertex $v$ of degree $4k$ adjacent to the vertices of an oriented cycle $v_{0,1}, v_{1,2}, \ldots, v_{k-1,0}, v_{1,3}, v_{2,4}, \ldots, v_{k-1,2}, v_{0,4}, \ldots, v_{0,2}, v_{k-1,1}, v_{k-2,3}, \ldots, v_{0,3}$ where each edge of the cycle is oriented forward. Intuitively, this oriented cycle will be embedded clockwise to encode the constraints induced by a matrix $M_i$ representing $V_i$ when its left-to-right order of columns is $\pi_i$. More precisely, a wheel replacement of cluster $V_i$ consistent with $\pi_i$ is the clustered graph obtained as follows: (i) remove $V_i$ and all the inter-cluster edges incident to $V_i$; (ii) insert the wheel $W_i$ of $V_i$ consistent with $\pi_i$; and (iii) for each inter-cluster edge $e = (u,v_j)$, with $v_j \in V_i$, insert edge $(u,v_{j,\pi_i(e)})$ incident to $W_i$. We call edge $(u,v_{j,\pi_i(e)})$ the image of edge $e = (u,v_j)$.

Let $G = (V,E,C,\Phi,\Pi)$ be a clustered graph with side assignment $\Phi$ where $\Pi$ is a set of permutations $\{\pi_1, \pi_2, \ldots, \pi_{|C|}\}$, one for each cluster $V_i$ (with $i = 1, \ldots, |C|\}$. We call $\Pi$ the permutation assignment of $G$ and we say that $G$ is NodeTrix planar with side assignment $\Phi$ and permutation assignment $\Pi$ if $G$ admits a NodeTrix planar representation with side assignment $\Phi$ where for each matrix $M_i$ the permutation of its columns is $\pi_i$. The wheel reduction of $G$ consistent with $\Pi$ is the graph obtained by performing a wheel replacement of $V_i \in C$ consistent with $\pi_i$ for each $i = 1, \ldots, |C|$. 

**Theorem 1.** Let $G = (V,E,C,\Phi,\Pi)$ be a clustered graph with side assignment $\Phi$ and permutation assignment $\Pi$. $G$ is NodeTrix planar if and only if the planar wheel reduction of $G$ admits a planar embedding where the external oriented cycle of each wheel $W_i$ is embedded clockwise.

Fig. 1(a) and Fig. 1(b) show respectively a NodeTrix planar representation of a clustered graph $G$ and the corresponding wheel reduction with its planar embedding.

Based on Theorem 1 we can test the graph $G = (V,E,C,\Phi)$ for NodeTrix planarity by exploring the space of the possible permutation sets $\Pi$ and corresponding wheel reductions in search of a NodeTrix planar $G = (V,E,C,\Phi,\Pi)$. Note that, if the maximum size of a cluster is given as a parameter $k$, every cluster $V_i$ can be replaced by $k!$ wheel graphs, one for each possible permutation of the vertices of $V_i$. In order to test planarity, for any such wheel replacement $W_i$, the cyclic order of the inter-cluster edges incident to the same vertex of $W_i$ can be arbitrarily permuted. While each wheel reduction yields an instance of constrained planarity testing that can be solved with the linear-time algorithm.
described in [6], a brute-force approach that repeats this algorithm on each possible wheel reduction may lead to testing planarity on $|C|^k$ different instances. Instead, for each visited node $\mu$ of the decomposition tree $T$ we compute a succinct description of the possible NodeTrix planar representations of the subgraph $G_\mu$ of $G$ represented by the subtree of $T$ rooted at $\mu$. This is done by storing for the poles of $\mu$ those pairs of wheel graphs that are compatible with a NodeTrix planar representation of $G_\mu$. How to efficiently compute such a succinct description will be the subject of the next sections.

4 Testing NodeTrix Planarity for Partial 2-Trees

In this section we prove that NodeTrix planarity testing with fixed sides can be solved in polynomial time for a clustered graph $G = (V, E, C, \Phi)$ when the maximum size of any cluster of $C$ is bounded by a constant and the frame graph is a partial 2-tree. This contrasts with the NP-hardness of NodeTrix planarity testing with fixed sides proved in [3] in the case where the size of the clusters is unbounded.

We first study the case of a clustered graph whose frame graph is a series-parallel graph, i.e., it is biconnected and its SPQR decomposition tree only has Q-, P-, and S-nodes. We then consider the case of partial 2-trees, i.e., graphs whose biconnected components are series-parallel.

4.1 Series-Parallel Frame Graphs

In this section we prove that NodeTrix planarity testing with fixed sides can be solved in $O(k^{3k+\frac{3}{2}} \cdot n^2)$ time for clustered graphs whose frame graphs are series-parallel and have cluster size at most $k$. 

---

**Fig. 1.** (a) A NodeTrix planar representation of a clustered graph. (b) The planar embedding of the corresponding wheel reduction. (c) Labeling of the vertices of $W_{\mu}$; the complete internal and external sequences are highlighted.
Let \( G = (V, E, C, \Phi) \) be a series-parallel clustered graph with side assignment \( \Phi \) and let \( F \) be its frame graph. Let \( T \) be the SPQ decomposition tree of \( F \) rooted at any Q-node. To simplify the description and without loss of generality, we assume that every S-node of \( T \) has exactly two children. Let \( \mu \) be a node of \( T \), and let \( s_\mu, t_\mu \) be the poles of \( \mu \). Consider the pertinent graph \( F_\mu \) represented by the subtree of \( T \) rooted at \( \mu \) and let \( v_\mu \) be a pole of \( \mu \) (\( v_\mu \in \{s_\mu, t_\mu\} \)). Pole \( v_\mu \) in the frame graph \( F \) may correspond to a non-trivial cluster \( V_i \) of \( C \). In this case, we call \( v_\mu \) a non-trivial pole of \( \mu \) and cluster \( V_i \) the pertinent cluster of \( v_\mu \).

The edges of \( F_\mu \) incident to \( v_\mu \) are the intra-component edges of \( v_\mu \). The other edges of \( F \) incident to \( v_\mu \) are the extra-component edges of \( v_\mu \). Each intra-component (extra-component) edge of \( v_\mu \) corresponds to an inter-cluster edge \( e' \) of \( G \) incident to one vertex of the pertinent cluster \( V_\mu \) of \( v_\mu \). We call \( e' \) an intra-component edge (extra-component edge) of \( V_\mu \). We associate \( k! \) wheel graphs to each non-trivial pole \( v_\mu \) of \( \mu \). Each of them is a wheel replacement of the pertinent cluster of \( v_\mu \), consistent with one of the \( k! \) permutations of its vertices.

Let \( v_\mu \) be a non-trivial pole of \( \mu \), let \( V_\mu \) be the pertinent cluster of \( v_\mu \), let \( \pi_\mu \) be a permutation of the vertices of \( V_\mu \), and let \( W_\mu \) be the wheel replacement of \( V_\mu \) consistent with \( \pi_\mu \). Every edge \( e \) incident to \( W_\mu \) such that \( e \) is the image of an inter-cluster edge \( e' \) of \( G \) is labeled either \text{int} or \text{ext}, depending on whether \( e' \) is an intra-component or an extra-component edge of \( V_\mu \). A vertex \( w \) of the external cycle of \( W_\mu \) is assigned one label of the set \{\text{void}, \text{int}, \text{ext}, \text{int-ext}\} as follows. Vertex \( w \) is labeled \text{void} if no edge incident to \( w \) is the image of an inter-cluster edge. Vertex \( w \) is labeled \text{int} (resp. \text{ext}) if we have a label \text{int} (resp. \text{ext}) on every edge \( e \) incident to \( w \) such that \( e \) is the image of an inter-cluster edge. Otherwise, vertex \( w \) is labeled \text{int-ext}. See Fig. 1(c) for an example concerning the wheel \( W_{t_\mu} \) of Fig. 1(b) the dashed curve of Fig. 1(b) shows the subgraph of the wheel reduction corresponding to \( F_\mu \).

A clockwise sequence \( v_0, v_1, \ldots, v_j \) of vertices of the external cycle of \( W_\mu \) is an external sequence of pole \( v_\mu \) consistent with \( \pi_\mu \) if \( v_0 \) and \( v_j \) are labeled either \text{ext} or \text{int-ext} and all the other vertices of the sequence are labeled either \text{void} or \text{ext}. An external clockwise sequence of pole \( v_\mu \) is complete if it contains all the vertices of \( W_\mu \) that are labeled \text{ext} and \text{int-ext}. Note that a complete external sequence may contain many \text{void} vertices but no \text{int} vertex. Internal and complete internal sequences of pole \( v_\mu \) are defined analogously. Observe that a complete internal sequence and a complete external sequence of \( v_\mu \) may not exist when vertices labeled \text{int} and vertices labeled \text{ext} alternate more than twice when traversing clockwise the external cycle of \( W_\mu \), or when three vertices are labeled \text{int-ext}. A special case is when \( W_\mu \) has exactly two vertices \( w_1 \) and \( w_2 \) labeled \text{int-ext} and all other vertices are \text{void}. In this case, the clockwise sequence from \( w_1 \) to \( w_2 \) and the clockwise sequence from \( w_2 \) to \( w_1 \) are both complete internal and complete external sequences.

In order to test \( G = (V, E, C, \Phi) \) for NodeTrix planarity, we implicitly take into account all possible permutation assignments \( \Pi \) by considering, for each non-trivial pole \( w_\mu \) of each node \( \mu \) of \( T \), its \( k! \) possible wheels and by computing their complete internal and complete external sequences. We visit the SPQ
decomposition tree $T$ from the leaves to the root and equip each node $\mu$ of $T$ with information regarding the complete internal and complete external sequences of its non-trivial poles. Let $\mu$ be an internal node of $T$, let $v_\mu$ be a non-trivial pole of $\mu$, let $\pi_v_\mu$ be a permutation of the pertinent cluster $V_\mu$ of $v_\mu$, and let $W_\mu$ be the wheel of $V_\mu$ consistent with $\pi_v_\mu$. We denote as $ISeq(\mu, v_\mu, \pi_v_\mu)$ the complete internal sequence of $v_\mu$ consistent with $\pi_v_\mu$ in pole $\mu$ and as $ESeq(\mu, v_\mu, \pi_v_\mu)$ the complete external sequence of $v_\mu$ consistent with $\pi_v_\mu$ in pole $\mu$. We distinguish between the different types of nodes of $T$.

**Node $\mu$ is a Q-node.** Since $G$ is light, at most one of its poles is non-trivial. Let $e$ be an edge of $F$ that is the pertinent graph of $\mu$. One end-vertex of $e$ is the representative vertex in $F$ of the pertinent cluster of the non-trivial pole $v_\mu$. In fact, edge $e$ corresponds to an edge $e' = (u, z)$ of $G$ such that $u \in V_\mu$ and $z$ is a trivial cluster. The side assignment $\phi_{v_\mu}$ defines whether $e$ is incident to the top, bottom, left, or right copy $uW$ of $u$ in the wheel $W_\mu$ of $V_\mu$. For any possible permutation $\pi_v_\mu$ we have $ISeq(\mu, v_\mu, \pi_v_\mu) = uW$. If $uW$ is labeled int-ext, then $ESeq(\mu, v_\mu, \pi_v_\mu)$ is the external cycle of $W_\mu$ starting at $uW$ and ending at $uW$. Otherwise, traverse the external cycle of $W_\mu$ starting at $uW$ and following the direction of the edges; $ESeq(\mu, v_\mu, \pi_v_\mu)$ consists of all the encountered vertices from the first labeled ext to the last labeled ext.

**Node $\mu$ is a P-node.** Let $\nu_0, \nu_1, \ldots, \nu_{h-1}$ be the children of $\mu$. Observe that $v_\mu$ is a non-trivial pole also for the children $\nu_0, \nu_1, \ldots, \nu_{h-1}$ of $\mu$. We consider every permutation $\pi_v_\mu$ such that $\nu_0, \nu_1, \ldots, \nu_{h-1}$ have both a complete internal sequence and a complete external sequence compatible with $\pi_v_\mu$. The complete internal sequence of $v_\mu$ consistent with $\pi_v_\mu$ is the union of the complete internal sequences of the children $\nu_0, \nu_1, \ldots, \nu_{h-1}$, that is $ISeq(\mu, v_\mu, \pi_v_\mu) = \bigcup_{i=0}^{h-1} ISeq(\nu_i, v_\mu, \pi_v_\mu)$.

To determine the complete external sequence of $v_\mu$ consistent with $\pi_v_\mu$ we consider the intersection of the complete external sequences of the children of $\mu$. If this intersection consists of exactly one sequence of consecutive vertices, then $ESeq(\mu, v_\mu, \pi_v_\mu) = \cap_{i=0}^{h-1} ESeq(\nu_i, v_\mu, \pi_v_\mu)$. Otherwise (i.e., the intersection is empty or it consists of more than one sequence of consecutive vertices), $v_\mu$ does not have a complete external sequence consistent with $\pi_v_\mu$.

**Node $\mu$ is an S-node.** Let $\nu$ be the child of $\mu$ that shares the pole $v_\mu$ with $\mu$. We consider every permutation $\pi_v_\mu$ such that $\nu$ has both $ISeq(\nu, v_\mu, \pi_v_\mu)$ and $ESeq(\nu, v_\mu, \pi_v_\mu)$. The complete internal (external) sequence of $v_\mu$ consistent with $\pi_v_\mu$ is $ISeq(\mu, v_\mu, \pi_v_\mu) = ISeq(\nu, v_\mu, \pi_v_\mu)$ ($ESeq(\mu, v_\mu, \pi_v_\mu) = ESeq(\nu, v_\mu, \pi_v_\mu)$).

To test $G$ for NodeTrix planarity we execute a bottom-up traversal of $T$ and, for each node $\mu$ with poles $s_\mu$ and $t_\mu$, we check whether each possible pair $(\pi_{s_\mu}, \pi_{t_\mu})$ induces complete internal and external sequences for $s_\mu$ and $t_\mu$ that are ‘compatible’ with a planar embedding of the wheel reduction of $G$. If this is the case, by Theorem 1, $G$ is NodeTrix planar, otherwise we reject $G$.

More formally, let $\pi_{s_\mu}$ ($\pi_{t_\mu}$, respectively) be a permutation such that $s_\mu$ ($t_\mu$, respectively) has both a complete internal sequence and a complete external sequence compatible with $\pi_{s_\mu}$ ($\pi_{t_\mu}$, respectively). We say that $(\pi_{s_\mu}, \pi_{t_\mu})$ is a
compatible pair of permutations for $\mu$ if either one of the poles is a trivial pole or one of the following cases applies.

**Node $\mu$ is a Q-node.** In this case all $k!$ possible pairs of permutations for $s_\mu$ or $t_\mu$ (recall that only one of them is non-trivial) are compatible for $\mu$.

**Node $\mu$ is a P-node.** Let $\nu_0, \nu_1, \ldots, \nu_{h-1}$ be the children of $\mu$. Consider a pair of permutations $(\pi_s, \pi_t)$; we recall that, for $i = 0, \ldots, h-1$, each $\nu_i$ has poles $s_\mu$ and $t_\mu$. A first condition for pair $(\pi_s, \pi_t)$ to be a compatible pair for $\mu$ is that $(\pi_s, \pi_t)$ is also a compatible pair for $\nu_i$, with $i = 0, \ldots, h-1$. A second condition asks that the pair $(\pi_s, \pi_t)$ defines opposite orders on the poles of $\mu$. Namely, let $W_\mu^s$ (resp., $W_\mu^t$) be the wheel of $V_s$ (resp., $V_t$) consistent with $\pi_s$ (resp., $\pi_t$).

Traversing clockwise the external cycle of $W_\mu^s$ starting from the first vertex of $ISeq(\mu, s_\mu, \pi_s)$, let $ISeq(\nu_0, s_\mu, \pi_s), ISeq(\nu_1, s_\mu, \pi_s), \ldots, ISeq(\nu_{h-1}, s_\mu, \pi_s)$ be the order by which the internal sequences are encountered. Pair $(\pi_s, \pi_t)$ defines opposite orders on the poles of $\mu$ if, traversing clockwise the external cycle of $W_\mu^t$ starting from the first vertex of $ISeq(\mu, t_\mu, \pi_t)$, the order by which we encounter the internal sequences of $\nu_0, \nu_1, \ldots, \nu_{h-1}$ is the opposite one, i.e., the order is $ISeq(\nu_{h-1}, t_\mu, \pi_t), ISeq(\nu_{h-2}, t_\mu, \pi_t), \ldots, ISeq(\nu_0, t_\mu, \pi_t)$.

**Node $\mu$ is an S-node.** Let $\nu_0$ and $\nu_1$ be the children of $\mu$ such that $s_0 = s_\mu$, $t_0 = s_\mu$, and $t_1 = t_\mu$. A pair $(\pi_s, \pi_t)$ is a compatible pair for $\mu$ if there exists a permutation $\pi_{t_0}$ such that the pair $(\pi_s, \pi_{t_0})$ is compatible for $\nu_0$ and the pair $(\pi_s, \pi_{t_0})$ is compatible for $\nu_1$.

Fig. 2 suggests that a NodeTrix planar representation of a clustered graph $G$ defines a permutation assignment $\Pi$ such that, for every node $\mu$ of $T$, pair $(\pi_s, \pi_t)$ is a compatible pair for $\mu$.

**Lemma 1.** Let $G = (V, E, C, \Phi)$ be a clustered graph with side assignment $\Phi$ and let $T$ be the SPQ decomposition tree of the frame graph of $G$. Graph $G$ is NodeTrix planar if and only if there exists a permutation assignment $\Pi$ such that, for every node $\mu$ of $T$ with poles $s_\mu$ and $t_\mu$, we have that permutation $\pi_s \in \Pi$ and permutation $\pi_t \in \Pi$ form a compatible pair of permutations for $\mu$.

**Lemma 2.** Let $G = (V, E, C, \Phi)$ be a series-parallel clustered graph with side assignment $\Phi$. Let $k$ be the maximum size of any cluster in $C$ and let $n$ be the cardinality of $V$. There exists an $O(k^{3k+\frac{3}{2}} \cdot n^3)$-time algorithm that tests whether $G$ is NodeTrix planar with side assignment $\Phi$ and if so, it computes a NodeTrix planar representation of $G$ consistent with $\Phi$.

**Proof.** Let $F$ be the frame graph of $G$; for any possible choice of an edge $e$ of $F$ we repeat the following procedure. We construct the SPQ decomposition tree of $G$ rooted at the Q-node whose pertinent graph is $e$. We visit $T$ from the leaves to the root and test whether $G$ has a permutation assignment $\Pi$ such that $G = (V, E, C, \Phi, \Pi)$ is NodeTrix planar. We first equip each non-trivial pole $v_\mu$ of every node $\mu$ of $T$ with its possible complete internal and complete external sequences. The maximum number of complete internal sequences of $v_\mu$ is $k!$. The same is true for the complete external sequences. If each complete (internal or
external) sequence of pole $\mu$ is encoded by means of its first and last vertex in the clockwise order around $W_\mu$, then each complete internal or external sequence needs constant space. It follows that the intersection or the union of two complete internal or external sequences can be computed in constant time. Therefore, all complete internal and external sequences for each non-trivial pole of $T$ can be computed in $O(k!)$ time. Hence, the whole bottom-up traversal to equip all non-trivial poles with every possible complete internal/external sequence can be executed in $O(k! \cdot n)$ time. We now test whether there exists a permutation assignment $\Pi$ such that any node $\mu$ of $T$ has a compatible pair of permutations. To this aim, we look at the complete internal and external sequences for the pair of poles of the children of $\mu$. For each pair $(\pi_s, \pi_t)$ of permutations of the poles of $\mu$ we equip $\mu$ with the information about whether such pair is compatible for $\mu$. This requires $O(k^2)$ space. If $\mu$ is a Q-node, every pair of permutations $(\pi_s, \pi_t)$ is compatible for $\mu$. It follows that all compatible pairs for $\mu$ can be computed in $O(k!)$ time (recall that one between $s_\mu$ and $t_\mu$ is non-trivial) and, hence, in $O(k! \cdot n)$ time for all the Q-nodes of $T$. If $\mu$ is a P-node with children $\nu_0, \nu_1, \ldots, \nu_{h-1}, \pi_s, \pi_t$ is one of the permutations that equip $s_\mu$, and $\pi_s, \pi_t$ is one of the permutations that equip $t_\mu$, testing whether the pair $(\pi_s, \pi_t)$ is a compatible pair for $\mu$ can be executed in $O(h)$ time. It follows that all compatible pairs for $\mu$ can be computed in $O(k!^2 \cdot h)$ time and, hence, in $O(k!^2 \cdot n)$ time for all P-nodes of $T$. If $\mu$ is an S-node with children $\nu_0$ and $\nu_1$, $\pi_s, \pi_t$ is one of the permutations that equip $s_\mu$, and $\pi_s, \pi_t$ is one of the permutations that equip $t_\mu$, testing whether the pair $(\pi_s, \pi_t)$ is a compatible pair for $\mu$ can be executed in $O(k!)$ time, corresponding to choosing all possible permutations for the pole shared between $\nu_0$ and $\nu_1$. It follows that all compatible pairs for $\mu$ can be computed in $O(k!^3)$ time and, hence, in $O(k!^3 \cdot n)$ time for all S-nodes of $T$. 

Fig. 2. (a) A NodeTrix planar representation $G = (V, E, \Phi)$. (b) $G$ induces a permutation assignment and a planar embedding of a wheel reduction of $G$; the complete internal and external sequences for a pair of poles are also highlighted.
In conclusion, the time complexity of a bottom-up visit of $T$ rooted at $e$ is $O(k!^3 \cdot n)$. By rooting $T$ at all possible Q-nodes, we obtain an overall time complexity of $O(k!^3 \cdot n^3)$. By Stirling’s approximation, $k! \sim \sqrt{2\pi k} \left(\frac{k}{e}\right)^k$ and thus a series-parallel clustered graph $G$ with $n$ vertices, side assignment $\Phi$, and maximum cluster size $k$ can be tested for NodeTrix planarity in $O(n^3 \cdot \ln n)$ time. Note that the compatible pair of permutations stored at each node $\mu$ of $T$ implicitly define a planar embedding of a wheel reduction of $G$. It can be shown that it is possible to construct a NodeTrix planar representation of $G$ in time proportional to the number of edges of $G$, which is $O(n \cdot k)$ [5]. The statement of the lemma follows.

4.2 Partial 2-Trees

We now consider clustered graphs whose cluster size is at most $k$ and such that their frame graph is a partial 2-tree, i.e. it is a planar graph whose biconnected components are series-parallel. We handle this case by decomposing the frame graph into its blocks and we store them into a block-cut-vertex tree. The following theorem generalizes the result of Lemma 2.

Theorem 2. Let $G = (V, E, C, \Phi)$ be a partial 2-tree clustered graph with side assignment $\Phi$. Let $k$ be the maximum size of any cluster in $C$ and let $n$ be the cardinality of $V$. There exists an $O(n^3 \ln n)$-time algorithm that tests whether $G$ is NodeTrix planar with side assignment $\Phi$ and if so, it computes a NodeTrix planar representation of $G$ consistent with $\Phi$.

5 General Planar Frame Graphs

In this section we study the problem of extending Theorem 2 to planar frame graphs that may not be partial 2-trees. We prove that NodeTrix planarity testing with fixed sides can be solved in polynomial time for maximum cluster size $k = 2$. However, the problem becomes NP-complete with fixed sides for $k \geq 3$ and it remains NP-complete even in the free sides scenario for $k \geq 5$.

Every block of the frame graph can be decomposed into its triconnected components by means of an SPQR decomposition tree. For each block, we adopt the same approach as for series-parallel graphs and look for a permutation assignment $\Pi$ such that, for every pair of poles $s_\mu$ and $t_\mu$, $(\pi_{s_\mu}, \pi_{t_\mu})$ forms a compatible pair for $\mu$ when $\mu$ is either a Q-node, a P-node, or an S-node. We extend the definition of compatible pairs of permutations for an R-node as follows.

Let $G = (V, E, C, \Phi)$ be a clustered graph with side assignment $\Phi$, let $F$ be the frame graph of $G$, and let $T$ be the SPQR decomposition tree of $F$. Let $\mu$ be an R-node of $T$ with poles $s_\mu$ and $t_\mu$. A pair of permutations $(\pi_{s_\mu}, \pi_{t_\mu})$ forms a compatible pair for $\mu$ if there exists a planar embedding of the skeleton $\text{skel}(\mu)$ of $\mu$ for which the following conditions hold: (i) For each vertex $v$ of $\text{skel}(\mu)$, let $e_0, e_1, \ldots, e_{h-1}$ be the virtual edges of $\text{skel}(\mu)$ incident to $v$ in clockwise order around $v$. Each such edge $e_i$ is associated with a child $\nu_i$ of
There exists a permutation \( \pi_v \) such that the complete internal sequences \( ISeq(\nu_0, v, \pi_v), ISeq(\nu_1, v, \pi_v), \ldots, ISeq(\nu_{h−1}, v, \pi_v) \) appear in this clockwise order around \( v \). (ii) Every vertex \( v \) of \( \text{skel}(\mu) \) can be assigned a permutation \( \pi_v \) such that:

- \( \pi_v = \pi_{s\mu} \) if \( v = s\mu \) and \( \pi_v = \pi_{t\mu} \) if \( v = t\mu \), and
- for each virtual edge \( e = (u, v) \) in \( \text{skel}(\mu) \) that corresponds to a child \( \nu \) of \( \mu \), the permutation pair \( (\pi_u, \pi_v) \) is compatible for \( \nu \).

Observe that, in the case of maximum cluster size \( k = 2 \), the possible permutations of the induced cluster \( V_v \) of a vertex \( v \) of \( \text{skel}(\mu) \) are exactly two, denoted by \( \pi_v^+ \) and \( \pi_v^- \). In order to test whether \( (\pi_{s\mu}, \pi_{t\mu}) \) forms a compatible pair for \( \mu \), we perform a traversal of \( \text{skel}(\mu) \) starting at \( s\mu \). Permutation \( \pi_{s\mu} \) and the clockwise order of the edges incident to \( s\mu \) can impose to choose only one of the two permutations \( \pi_w^+ \) o \( \pi_w^- \) available for each vertex \( w \) adjacent to \( s\mu \) and corresponding to a non-trivial cluster of \( G \). Each such \( w \) and its incident edges, in turn, propagate constraints on the possible permutations to their neighbors, till \( t\mu \) is reached. Therefore, testing whether \( \pi_{s\mu} \) and \( \pi_{t\mu} \) form a compatible pair for \( \mu \) can be reduced to a suitable problem of labeling the edges and vertices of \( \text{skel}(\mu) \) and verifying that at the end \( s\mu \) and \( t\mu \) are labeled with \( \pi_{t\mu} \) and \( \pi_{s\mu} \).

Theorem 3. Let \( G = (V, E, C, \Phi) \) be an \( n \)-vertex clustered graph with side assignment \( \Phi \) such that the maximum size of any cluster in \( C \) is two. There exists an \( O(n^3) \)-time algorithm that tests whether \( G \) is NodeTrix planar with the given side assignment and if so, computes a NodeTrix planar representation of \( G \) consistent with \( \Phi \).

The proof of the following theorem is based on a reduction from (non-planar) NAE3SAT.

Theorem 4. NodeTrix planarity testing with fixed sides and cluster size at most \( k \) is NP-complete for any \( k \geq 3 \).

Now, we extend the above hardness result to the free sides model and show that NodeTrix planarity testing remains NP-complete when the maximum cluster dimension is larger than four. This is done by proving that NAE3SAT is NP-complete even for triconnected Boolean formulas, which may be a result of independent interest.

Theorem 5. NAE3SAT is NP-complete for triconnected Boolean formulas.

Theorem 6. NodeTrix planarity testing with free sides and cluster size at most \( k \) is NP-complete for any \( k \geq 5 \).

6 Open Problems

We conclude the paper by listing some open problems that, in our opinion, are worth investigating. (i) Study the complexity of NodeTrix planarity testing in the free sides scenario for values of \( k \) between 2 and 5. (ii) Study families of clustered graphs for which NodeTrix planarity testing is fixed parameter tractable in the free sides scenario. (iii) Determine whether the time complexity of the algorithms in Theorems 2 and 3 can be improved.
References

1. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Rutter, I.: Intersection-link representations of graphs. In: Di Giacomo, E., Lubiw, A. (eds.) Proceedings of 23rd International Symposium on Graph Drawing and Network Visualization (GD ’15). Lecture Notes in Computer Science, vol. 9411, pp. 217–230 (2015)
2. Batagelj, V., Brandenburg, F., Didimo, W., Liotta, G., Palladino, P., Patrignani, M.: Visual analysis of large graphs using (X,Y)-clustering and hybrid visualizations. IEEE Trans. Vis. Comput. Graph. 17(11), 1587–1598 (2011)
3. Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M.: Computing NodeTrix representations of clustered graphs. In: Nöllenburg, M., Hu, Y. (eds.) Graph Drawing and Network Visualization (GD ’16). Lecture Notes in Computer Science, vol. 9801, pp. 107–120 (2016)
4. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing. Prentice Hall, Upper Saddle River, NJ (1999)
5. Di Giacomo, E., Liotta, G., Patrignani, M., Tappini, A.: Planar k-NodeTrix Graphs - a new family of beyond planar graphs. In: Frati, F., Ma, K. (eds.) Graph Drawing and Network Visualization (GD 2017). LNCS, Springer, to appear
6. Gutwenger, C., Klein, K., Mutzel, P.: Planarity testing and optimal edge insertion with embedding constraints. J. Graph Algorithms Appl. 12(1), 73–95 (2008)
7. Harary, F.: Graph Theory. Addison-Wesley Series in Mathematics, Addison Wesley (1969)
8. Henry, N., Fekete, J., McGuñin, M.J.: NodeTrix: a hybrid visualization of social networks. IEEE Trans. Vis. Comput. Graph. 13(6), 1302–1309 (2007)
9. Kratochvíl, J.: A special planar satisfiability problem and a consequence of its NP-completeness. Discrete Applied Mathematics 52(3), 233 – 252 (1994)
10. Moret, B.M.E.: Planar NAE3SAT is in P. SIGACT News 19(2), 51–54 (1988)
11. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing. pp. 216-226 (1978)
A Omitted Proofs

A.1 Proof of Theorem 1

Theorem 1. Let $G = (V, E, C, \Phi, \Pi)$ be a clustered graph with side assignment $\Phi$ and permutation assignment $\Pi$. $G$ is NodeTrix planar if and only if the planar wheel reduction of $G$ admits a planar embedding where the external oriented cycle of each wheel $W_i$ is embedded clockwise.

Proof. If $G$ is NodeTrix planar, we construct a planar embedding of a wheel reduction of $G$ where the external oriented cycle of each wheel is embedded clockwise as follows. Let $\Gamma$ be a NodeTrix planar representation of $G$. We replace each matrix $M_i$ representing a cluster $V_i \in C$ with the wheel $W_i$ of $V_i$ consistent with the permutation $\pi_i \in \Pi$ of $V_i$. Also $W_i$ is embedded in such a way that a forward traversal of its external cycle is a clockwise traversal of the cycle. Every inter-cluster edge $e = (u, v_j)$, with $v_j \in V_i$, is incident to the vertex $v_{j,\phi(e)}$ of the wheel $W_i$. Also, for all $j = 0, 1, \ldots, k - 1$ and for all $x \in \{T, B, L, R\}$, the cyclic order of the inter-cluster edges incident to $v_{j,x}$ in $W_i$ is the same as the cyclic order of the inter-cluster edges incident to $p_{v_{j,x}}$ in $M_i$. It is immediate to see that, since no two inter-cluster edges cross in $\Gamma$, no two edges cross in the constructed embedding of the wheel reduction of $G$.

Conversely, suppose that we are given a planar embedding of the wheel reduction of $G$ where the external oriented cycle of each wheel is embedded clockwise. We show how to construct a NodeTrix planar representation of $G$. For each wheel $W_i$ we remove the center vertex of the wheel and insert a matrix $M_i$ inside the created face. We now morph every vertex $v_{j,\phi(e)}$ of the external cycle of $W_i$ to point $p_{v_{j,\phi(e)}}$ in $M_i$ and maintain around $p_{v_{j,\phi(e)}}$ the cyclic order of the inter-cluster edges incident to $v_{j,\phi(e)}$ in the planar embedding of the wheel reduction.

\[\square\]

A.2 Proof of Lemma 1

Lemma 1. Let $G = (V, E, C, \Phi)$ be a clustered graph with side assignment $\Phi$ and let $T$ be the SPQ decomposition tree of the frame graph of $G$. Graph $G$ is NodeTrix planar if and only if there exists a permutation assignment $\Pi$ such that, for every node $\mu$ of $T$ with poles $s_\mu$ and $t_\mu$, we have that permutation $\pi_{s_\mu} \in \Pi$ and permutation $\pi_{t_\mu} \in \Pi$ form a compatible pair of permutations for $\mu$.

Proof. We prove first that, if $G = (V, E, C, \Phi)$ is NodeTrix planar, then there exists a permutation assignment $\Pi$ such that, for every node $\mu$ of $T$ with poles $s_\mu$ and $t_\mu$, the pair $(\pi_{s_\mu}, \pi_{t_\mu})$ is compatible for $\mu$.

Let $\Gamma$ be a NodeTrix planar representation of $G$ with side assignment $\Phi$ and let $M_0, M_1, \ldots, M_h$ be the matrices representing the non-trivial clusters of $G$. For each matrix $M_i$ ($i = 0, \ldots, h - 1$) of $\Gamma$, let $\pi_i = \pi_0, \pi_1, \ldots, \pi_{k-1}$ be the left to right order of the columns of $M_i$. We replace $M_i$ with a wheel $W_i$ consisting of a vertex $v_i$ of degree $4k$ adjacent to all vertices of a cycle $v_{0,T}, v_{1,T}, \ldots, v_{k-1,T}, v_{0,B}, v_{1,B}, \ldots, v_{k-1,B}, v_{k-1, R}, v_{k-2, B}, \ldots, v_{k-1, B}, v_{k-1, R}, v_{k-2, L}, \ldots, v_{0, L}$. For all $j = 0, 1, \ldots, k - 1$ and for all $x \in \{T, B, L, R\}$, vertex $v_{j, x}$ is drawn at the point $p_{v_{j, x}}$, that represents the attachment of the inter-cluster edges incident to vertex $v_j$ on the side $x$ of matrix $M_i$. The edges of the external cycle of $W_i$ are drawn along the external boundary of $M_i$. Every inter-cluster edge $e = (u, v_j)$, with $v_j \in M_i$, is incident to the vertex $v_{j,\phi(e)}$ of the wheel $W_i$. Also, for all $j = 0, 1, \ldots, k - 1$ and for all $x \in \{T, B, L, R\}$, the cyclic order of the inter-cluster edges incident to $v_{j, x}$ in $W_i$ is the same as the cyclic order of the inter-cluster edges incident to $p_{v_{j, x}}$ in $M_i$. It is straightforward to verify that the computed drawing defines a planar embedding for the wheel reduction of $G = (V, E, C, \Phi)$ consistent with $\Pi = \{\pi_1, \pi_2, \ldots, \pi_{|C|}\}$. From the planarity of the wheel reduction of $G$ it follows that each non-trivial pole $v_\mu$ of the frame graph $F$ has a complete internal and a complete external sequence consistent with $\pi_{s_\mu}$ and that for every node $\mu$ of the SPQ decomposition tree of $F$ having poles $s_\mu$ and $t_\mu$, the pair $(\pi_{s_\mu}, \pi_{t_\mu})$, with $\pi_{s_\mu}, \pi_{t_\mu} \in \Pi$, is compatible for $\mu$. An example of the above described procedure is illustrated in Fig. 2.

We now show that, if there exists a permutation assignment $\Pi$ such that for every node $\mu$ of $T$ with poles $s_\mu$ and $t_\mu$ we have that permutation pair $(\pi_{s_\mu}, \pi_{t_\mu})$ is compatible for $\mu$, then $G = (V, E, C, \Phi)$ is NodeTrix planar with side assignment $\Phi$. We construct a planar embedding of the wheel reduction of $G$ consistent with $\Pi$ such that all external cycles of the wheels are embedded clockwise which, by Theorem 1, implies that $G$ is NodeTrix planar. Let $W_{s_\mu}$ and $W_{t_\mu}$ be the two wheels consistent with $\pi_{s_\mu}$ and $\pi_{t_\mu}$ of $s_\mu$ and $t_\mu$, respectively. We visit $T$ from the leaves to the root and incrementally construct the desired planar embedding of the wheel reduction of $G$.

If the visited node $\mu$ is a Q-node, at most one of its poles is non-trivial because $G$ is light; assume, without loss of generality, that the non-trivial pole of $\mu$ is $s_\mu$ and let $V_{s_\mu}$ be the cluster of $G$ represented by $s_\mu$ in the frame graph of $G$. We embed the wheel $W_{s_\mu}$ of $V_{s_\mu}$ consistent with $\pi_{s_\mu} \in \Pi$ such that, when traversing the edges of the external cycle of $W_{s_\mu}$ in the forward direction, the cycle is traversed clockwise. We embed $t_\mu$ in the external face of $W_{s_\mu}$ and planarly connect the top, bottom, left, or right copy of its end-vertex on $W_{s_\mu}$ as specified by $\Phi$.  

\[\square\]
Suppose now μ is an S-node and let ν₀ and ν₁ be the children of μ such that s_{ν₀} = s_μ, t_{ν₀} = s_{ν₁}, and t_{ν₁} = t_μ. The planar embedding of the wheel reduction at node μ is obtained by composing the planar embedding of the wheel reduction at node ν₀ with the planar embedding of the wheel reduction at ν₁. This is done by identifying the planar embedding of the wheel W_{s_{ν₀}} consistent with π_{s_{ν₀}} with the planar embedding of the wheel W_{s_{ν₁}} consistent with π_{s_{ν₁}}. Note that this is possible because π_{s_{ν₀}} is the same as π_{s_{ν₁}}, (π_{s_{ν₁}}, π_{t_{ν₁}}) is a compatible pair for ν₁, and (π_{s_{ν₀}}, π_{t_{ν₀}}) is a compatible pair for ν₀.

Finally, assume μ is a P-node and let ν₀, ν₁, ..., ν_{h−1} be the children of μ. Similar to the case of the S-node, the planar embedding of the wheel reduction at node μ is obtained by composing the planar embeddings of the wheel reduction at nodes ν₀, ν₁, ..., ν_{h−1}. Since pair (π_{s_{ν₁}}, π_{t_{ν₁}}) is compatible for μ, it defines opposite orders on the poles of μ. These opposite circular orders correspond to a planar embedding of the wheel reduction at μ obtained by combining the planarly embedded wheel reductions at its children ν₀, ν₁, ..., ν_{h−1}. It follows that G = (V, E, C, Φ) is NodeTrix planar with permutation assignment Π.

A.3 Proof of Theorem 2

**Theorem 2.** Let G = (V, E, C, Φ) be a partial 2-tree clustered graph with side assignment Φ. Let k be the maximum size of any cluster in C and let n be the cardinality of V. There exists an \(O(k^{3k + \frac{3}{2}} \cdot n^3)\)-time algorithm that tests whether G is NodeTrix planar with side assignment Φ and if so, it computes a NodeTrix planar representation of G consistent with Φ.

**Proof.** We compute a block-cut-vertex tree T_{bcv} of the frame graph of G, we root it at a block B_{root} and visit T_{bcv} bottom-up. Let B_{t} be the currently visited block and let c be the parent cut-vertex of B_{t} in T_{bcv}. We execute the testing algorithm of Lemma 2 by rooting the SPQ decomposition tree of B_{t} only at those Q-nodes that have c as one of their poles. If the test fails at any block of T_{bcv}, we conclude that G is not NodeTrix planar with the given side assignment for the chosen root B_{root} of T_{bcv} and we repeat the test by rooting T_{bcv} at a different block.

Otherwise, we test whether, among the permutation assignments computed for the blocks B₀, B₁, ..., B_{h−1} that are children of a same cut-vertex c, there exists a set \{Π₀, Π₁, ..., Π_{h−1}\} such that: (i) Π₀,c = Π₁,c = ... = Π_{h−1},c with π_{i,j} \in Π_{h−1}, for j = 0, 1, ..., h − 1, and (ii) the complete internal sequence of c for the block B_{t} and for the permutation assignment Π_{i,c} does not overlap with the complete internal sequence of c for the block B_{i} and the permutation assignment Π_{i,c}, with i ≠ j, i, j = 0, ..., h − 1. We equip c with all the permutations that pass this test. Let B' be the block that is the parent of c in T_{bcv}. When testing B' for NodeTrix planarity, we consider for c only the permutations that have been computed when processing blocks B₀, B₁, ..., B_{h−1} and check that the complete internal sequences of c in B_j, j = 0, ..., h − 1, do not intersect with the complete internal sequences of c in B'. Let \(n_B\) be the number of vertices of a block B. By using Lemma 2 the procedure described above can be executed in time \(O(k^{3k + \frac{3}{2}} \cdot n_B^3)\) per block. Therefore, a bottom-up visit of T_{bcv} can be computed in \(O(k^{3k + \frac{3}{2}} \cdot n^3)\) time for each root block. Since there are \(O(n)\) possible roots for T_{bcv}, it follows that the overall time complexity is \(O(k^{3k + \frac{3}{2}} \cdot n^3)\).

A.4 Proof of Theorem 3

The lemmas and the discussion in this section prove Theorem 3. For completeness we report the statement of the theorem.

**Theorem 3.** Let G = (V, E, C, Φ) be an n-vertex clustered graph with side assignment Φ such that the maximum size of any cluster in C is two. There exists an \(O(n^3)\)-time algorithm that tests whether G is NodeTrix planar with the given side assignment and if so, computes a NodeTrix planar representation of G consistent with Φ.

Let G = (V, E, C, Φ) be a clustered graph with side assignment Φ and let F be its frame graph. Assume first that the frame graph F of G is biconnected. Let T be the SPQR decomposition tree of F.

Analogously to the case of series-parallel frame graph described in Section 4.1, we traverse T from the leaves to the root and compute, for each node μ of T with poles s_μ and t_μ, all the pairs of permutations \((π_{s_μ}, π_{t_μ})\) compatible for μ (the formal definitions of pair of permutations compatible for a Q-node, an S-node, or a P-node can be found in Section 4.1) the formal definition of pair of permutations compatible for an R-node is in Section 5.

Differently from the algorithm for series-parallel frame graphs, we have to handle the case of an R-node μ, where the skeleton skel(μ) is a triconnected graph. In fact, the cases of Q-node, S-node and P-node can be handled exactly as described in Section 4.1 with an overall time complexity \(O(n^2)\) (Lemma 2).

Let μ be an R-node of T with poles s_μ and t_μ and with children ν₀, ν₁, ..., ν_{h−1}. Given a non-trivial vertex w of skel(μ), since the maximum cluster size is two, the possible permutations of V_w are exactly two, that we denote by
are also vertices and edges of $G$. Removing $G$ is easy to prove: Every solution for a coherent labeling.

Claim 1

Proof. A succinct instance

In any coherent labeling for $G^*$ cannot be selected for vertex $w$ if there exists a virtual edge $e = (w, v)$ of $skel(\mu)$ that does not have any label $(\pi^+_w, \pi^-_w)$, with $\pi^+_w, \pi^-_w \in \{\pi^+_w, \pi^-_w\}$.

Edge Coherence: In any coherent labeling for $G$, label $(\pi^+_w, \pi^-_w)$ cannot be selected for the virtual edge $e = (w, v)$ of $skel(\mu)$ if $w$ does not admit label $\pi^+_w$ or $v$ does not admit label $\pi^-_w$.

Hence, we can discard vertex and edge labels that do not satisfy the above properties. An instance of the COHERENT-LABELING-PROBLEM where each vertex label satisfies Property Vertex Coherence and each edge label satisfies Property Edge Coherence is said to be succinct. Observe that, each time a vertex label is discarded because of Property Vertex Coherence at vertex $w$, the labels of each virtual edge incident to $w$ have to be checked against Property Edge Coherence. Also, each time an edge label is discarded because of Property Edge Coherence, Property Vertex Coherence has to be checked for the labels of one of its incident vertices. Due to the fact that at most one label per vertex and at most three labels per edge can be discarded (otherwise the instance is not NodeTrix planar), an instance of COHERENT-LABELING-PROBLEM can be reduced to a succinct instance in time linear in the number of edges of $skel(\mu)$.

A succinct instance $G$ of the COHERENT-LABELING-PROBLEM is called reduced if each vertex of $G$ has exactly two labels and each edge of $G$ has exactly three labels. We use the following lemma to efficiently solve the problem.

Lemma 3. Let $G$ be an instance of the COHERENT-LABELING-PROBLEM. There exists a reduced instance $G^*$ such that $G$ admits a coherent labeling if and only if $G^*$ admits a coherent labeling.

Proof. Graph $G^*$ can be found through a sequence of transformations starting from the original instance $G$. The first transformation is described in the proof of the following claim.

Claim 1

Proof of claim. Let $w$ be a vertex of $G$ that has only one label and let $G^{-w}$ be the graph obtained from $G$ by removing $w$ and all its incident edges. We show that $G^{-w}$ admits a coherent labeling if and only if $G$ admits one. One direction is easy to prove: Every solution for $G$ is also a solution for $G^{-w}$, since all vertices and edges of $G^{-w}$ are also vertices and edges of $G$. We now prove the opposite direction. Suppose vertex $w$ has only label $\pi^+_w$ (the case of $\pi^-_w$ being analogous) and consider any edge $(v, w)$ incident to $w$. By Property Edge Coherence, $(v, w)$ cannot
have labels \((\pi^+_w, \pi^-_w)\) and \((\pi^+_v, \pi^-_v)\). It follows that edge \((v, w)\) may have: (1) label \((\pi^+_v, \pi^+_w)\); (2) label \((\pi^-_v, \pi^+_w)\); or (3) both labels \((\pi^+_v, \pi^+_w)\) and \((\pi^-_v, \pi^-_w)\).

In turn, by Property Vertex Coherence for the labels of vertex \(v\), we have the following three cases, respectively: (1) vertex \(v\) has only label \(\pi^+_v\); (2) vertex \(v\) has only label \(\pi^-_v\); and (3) vertex \(v\) has both labels \(\pi^+_v\) and \(\pi^-_v\).

Suppose \(G^{-w}\) admits a solution. You can easily obtain a solution for the original instance \(G\) by inserting \(w\) and applying it to a solution \(\pi\) for \(G^{-w}\). Select, for each edge \(e = (v, w)\) incident to \(w\), depending on the label \(\pi^+_v\) or \(\pi^-_v\) of \(v\), label \((\pi^+_v, \pi^+_w)\) or \((\pi^-_v, \pi^-_w)\), respectively. Depending on the three cases above, we prove that \(G\) admits a coherent labeling.

1. Since any solution of \(G^{-w}\) selects label \(\pi^+_v\) for vertex \(v\), selecting label \(\pi^+_w\) for vertex \(w\) and label \((\pi^+_v, \pi^+_w)\) for edge \((v, w)\) yields a solution for \(G\).
2. Since any solution of \(G^{-w}\) selects label \(\pi^-_v\) for vertex \(v\), selecting label \(\pi^+_w\) for vertex \(w\) and label \((\pi^-_v, \pi^+_w)\) for edge \((v, w)\) yields a solution for \(G\).
3. A solution of \(G^{-w}\) may select either label \(\pi^+_v\) (Case A) or label \(\pi^-_v\) (Case B) for vertex \(v\). Selecting label \(\pi^+_w\) for vertex \(w\) and label \((\pi^-_v, \pi^+_w)\) in Case A or \((\pi^-_v, \pi^+_w)\) in Case B for edge \((v, w)\) yields a solution for \(G\).

By repeating this procedure for every vertex of \(G\) that has only one label, an equivalent instance \(G^*\) such that all vertices have both labels is obtained. Finally, observe that, since the removal of edges from an instance of Coherent-Labeling-Problem does not disrupt the properties of vertex coherence and edge coherence of its labels, when applying the transformation described above we obtain a succinct instance \(G^*\) whenever the original instance \(G\) was succinct.

Claim \(\Box\) has several important consequences. In order to explicit them, we need some further notation. We say that an edge \((u, v)\) has label set \(x\), with \(x \in \{1, 2, \ldots, 16\}\), if \((u, v)\) has the following sets of labels:

| Label set | \((\pi^+_w, \pi^-_w)\) | \((\pi^+_v, \pi^-_v)\) | \((\pi^+_u, \pi^-_u)\) | \((\pi^-_w, \pi^-_v)\) |
|-----------|-----------------|-----------------|-----------------|-----------------|
| 1         | ×               | ×               | ×               | ×               |
| 2         |                  | ×               | ×               | ×               |
| 3         |                  |                  | ×               | ×               |
| 4         |                  |                  |                  | ×               |
| 5         |                  |                  |                  |                  |
| 6         |                  |                  |                  |                  |
| 7         |                  |                  |                  |                  |
| 8         |                  |                  |                  |                  |
| 9         |                  |                  |                  |                  |
| 10        |                  |                  |                  |                  |
| 11        |                  |                  |                  |                  |
| 12        |                  |                  |                  |                  |
| 13        |                  |                  |                  |                  |
| 14        |                  |                  |                  |                  |
| 15        |                  |                  |                  |                  |
| 16        |                  |                  |                  |                  |

An immediate consequence of Claim \(\Box\) is the following property.

**Claim 2** If each vertex \(w\) of an instance \(G\) of the Coherent-Labeling-Problem has both the labels \(\pi^+_w\) and \(\pi^-_w\), then edges have exclusively label sets in \(\{8, 9, 12, 13, 14, 15, 16\}\).

**Proof of claim.** The proof is based on the observation that the remaining label sets either miss \(\pi^+_w\) (label sets 1, 4, 5, and 11), or miss \(\pi^-_w\) (among others, label sets 2, 3, and 6), or miss \(\pi^+_v\) (among others, label set 10), or \(\pi^-_w\) (among others, label set 7). However, by Property Vertex Coherence, \(u\) and \(v\) cannot have both labels if there is some edge that does not allow both of them.

**Claim 3** Let \(G\) be a succinct instance of the Coherent-Labeling-Problem. There exists a succinct instance \(G^*\) such that no edge has label set 16 and such that \(G^*\) admits a coherent labeling if and only if \(G\) admits one. Further, a coherent labeling for \(G\) can be obtained from a coherent labeling for \(G^*\) in constant time.

**Proof of claim.** If an edge \((u, v)\) has all four labels \((\pi^+_u, \pi^-_u), (\pi^+_v, \pi^-_v), (\pi^-_u, \pi^-_v), (\pi^-_u, \pi^-_v)\), one could remove \((u, v)\), compute a coherent labeling of the obtained instance \(G^*\), and then insert back again \((u, v)\) with a suitable label depending on the pair of labels selected for its end-vertices.
Claim 4 Let $G$ be a succinct instance of the \textsc{Coherent-Labeling-Problem}. There exists a succinct instance $G^*$ such that no edge has label set 8 and such that $G^*$ admits a coherent labeling if and only if $G$ admits one. Further, a coherent labeling for $G$ can be obtained from a coherent labeling for $G^*$ in constant time.

Proof of claim. Suppose edge $(u, v)$ only has labels $(\pi_u^+, \pi_v^-)$ and $(\pi_u^-, \pi_v^+)$. Contract edge $(u, v)$ and merge vertices $u$ and $v$ into a new vertex $w$. Suppose that the obtained graph $G^*$ admits a coherent labeling. Expand the vertex $w$ back into $u$ and $v$. If $w$ has label $\pi_u^+$, select label $\pi_u^-$ for $u$, label $\pi_v^+$ for $v$, and label $(\pi_u^+, \pi_v^-)$ for edge $(u, v)$. Otherwise, select label $\pi_u^-$ for $u$, label $\pi_v^-$ for $v$, and label $(\pi_u^-, \pi_v^+)$ for $(u, v)$. It is easy to verify that the obtained selection of labels is a coherent labeling for $G$.

Claim 5 Let $G$ be a succinct instance of the \textsc{Coherent-Labeling-Problem}. There exists a succinct instance $G^*$ such that no edge has label set 9 and such that $G^*$ admits a coherent labeling if and only if $G$ admits one. Further, a coherent labeling for $G$ can be obtained from a coherent labeling for $G^*$ in constant time.

Proof of claim. Suppose edge $(u, v)$ only has labels $(\pi_u^+, \pi_v^-)$ and $(\pi_u^-, \pi_v^+)$. Replace $\pi_u^-$ with $\pi_v^+$ and $\pi_v^+$ with $\pi_u^-$ for the labels of $v$ and for all the labels of the edges incident to $v$. Observe that the instance where the labels have been renamed admits a coherent labeling if and only if instance $G$ admits one. In particular, edges with label set 9 (12, 13, 14, 15, respectively) have now label set 8 (13, 12, 15, 14, respectively). Hence, for edge $(u, v)$ Claim 4 applies, and the edge can be contracted and reinserted after a coherent labeling for the resulting instance $G^*$ is found.

Claims 1, 2, 3, 4, and 5 imply that we can assume that every vertex has both labels and every edge has a label set in $\{12, 13, 14, 15\}$. Hence, the obtained instance $G^*$ is a reduced instance of the \textsc{Coherent-Labeling-Problem} equivalent to instance $G$, which concludes the proof of Lemma 3.

Based on Lemma 3, we can assume that every vertex $w$ of $G$ has exactly two labels in $\{\pi_u^+, \pi_u^-\}$ and every edge $(u, v)$ of $G$ has exactly three labels in $\{(\pi_u^+, \pi_v^+), (\pi_u^+, \pi_v^-), (\pi_u^-, \pi_v^+), (\pi_u^-, \pi_v^-)\}$. Consider an edge $(u, v)$ whose missing label is $(\pi_u^+, \pi_v^+)$. This implies that, if the instance $G$ of the \textsc{Coherent-Labeling-Problem} has a coherent labeling in which label $\pi_u^+$ is selected for vertex $u$, then the label selected for vertex $v$ in this solution must be $\pi_v^-$. This consideration can be generalized as follows:

Property 2. Let $G$ be a reduced instance of the \textsc{Coherent-Labeling-Problem} and let $e = (u, v)$ be an edge of $G$ missing label $(\pi_u^+, \pi_v^+)$, with $\pi_u^+ \in \{\pi_u^+, \pi_u^-\}$ and $\pi_v^+ \in \{\pi_v^+, \pi_v^-\}$. In any coherent labeling of $G$, if the label selected for $u$ is $\pi_u^+$, then the label selected for $v$ is the label different from $\pi_v^+$. Analogously, if the label selected for $v$ is $\pi_v^+$, then the label selected for $u$ is the label different from $\pi_u^+$. Property 2 shows that selecting a label for a vertex may have consequences on the label that have to be selected for the neighbor vertices, and a local choice may propagate in the graph. However, Property 2 also implies the following property.

Property 3. If the missing label of edge $(u, v)$ is such that the selection of a label $\pi_u^+$ for vertex $u$, with $\pi_u^+ \in \{\pi_u^+, \pi_u^-\}$, has no implication on the selection of the label for $v$, then any selection of a label $\pi_v^+$ for vertex $v$, with $\pi_v^+ \in \{\pi_v^+, \pi_v^-\}$, is compatible with the selection of label $\pi_u^+$ for vertex $u$.

This gives rise to the following procedure to compute all pairs of permutations $(\pi_{s\mu}^+, \pi_{t\mu}^+)$ compatible for R-node $\mu$. Since $\text{skel}(\mu)$ has two possible planar embeddings, the procedure will be repeated twice, once for each embedding. Let $E_{\mu}$ be the current embedding of $\text{skel}(\mu)$.

- Build an instance $G$ of \textsc{Coherent-Labeling-Problem} (if a vertex or edge has zero labels, then $\mu$ does not admit any compatible pair when $E_{\mu}$ is the embedding chosen for $\text{skel}(\mu)$).
- Obtain a reduced instance $G^*$ of the \textsc{Coherent-Labeling-Problem} that admits a coherent labeling if and only if $G$ admits one. Let $s^*$ and $t^*$ be the two vertices of $G^*$ corresponding to the vertices $s_\mu$ and $t_\mu$ of the original instance $G$.
- Consider one by one the four possible permutation pairs of $\mu$. To decide whether a permutation pair $(\pi_{s\mu}^+, \pi_{t\mu}^+)$ is compatible for $\mu$, select for $s^*$ the label $\pi_{s\mu}^+$ corresponding to $\pi_{s\mu}^+$ and propagate the selection through $G$ according to Property 2. If a vertex $w$ is reached twice times, once by imposing the selection of label $\pi_w^+$ and once by imposing the selection of label $\pi_w^-$, then the pair $(\pi_{s\mu}^+, \pi_{t\mu}^+)$ is not compatible for $\mu$. Suppose this first propagation phase stops without any such contradiction. If a label for vertex $t^*$ has not been selected yet, start a second propagation by selecting for $t^*$ the label $\pi_{t\mu}^+$ corresponding to $\pi_{t\mu}^+$. While there are vertices for which no
label has been selected, start a new propagation phase selecting an arbitrary label for one of them. Property 3 guarantees that a vertex traversed during a propagation phase cannot be traversed in a successive propagation phase. Hence, the labels selected during one propagation phase cannot cause a contradiction with the labels selected in successive propagation phases. If no contradiction is found and if \( \pi^*_t \) is the selected label for \( t^* \), permutation pair \( \{ \pi^*_s, \pi^*_t \} \) is compatible for \( \mu \).

The proof of Theorem 3 is concluded by observing that each of the transformations described in the proof of Lemma 3 requires constant time and therefore the overall time complexity to compute all pairs of permutations \( \{ \pi^*_s, \pi^*_t \} \) is linear in the number of edges of \( \text{ske}(\mu) \). This observation, together with Lemma 2, implies that NodeTrix planarity with fixed sides can be tested in \( O(n^2) \) time on a clustered graph whose maximum cluster size is two and such that the frame graph is biconnected.

For the connected case we construct the block-cut-vertex tree \( T_{b,cv} \) of the frame graph of \( G \) and apply the same strategy described in the proof of Theorem 2. Namely, for every node \( \mu \) that is either a Q-, or an S-, or a P-node, we test whether there exists a compatible pair for \( \mu \) by means of the procedure in Lemma 2; if \( \mu \) is an R-node we use the strategy described above. This gives rise to an \( O(n^2) \)-time procedure for each possible choice of the root of \( T_{b,cv} \). Since there are \( O(n) \) such possible choices, we have an overall time complexity of \( O(n^3) \) if the frame graph is not biconnected.

The statement of Theorem 3 follows.

A.5 Proof of Theorem 4

Theorem 4. NodeTrix planarity testing with fixed sides and cluster size at most \( k \) is NP-complete for any \( k \geq 3 \).

Proof. NodeTrix planarity testing with fixed sides is trivially in NP. In fact, given a clustered graph \( G = (V, E, C, \Phi) \), all possible permutations assignments \( \Pi \) could be non-deterministically computed and the problem of deciding whether a clustered graph \( G = (V, E, C, \Phi, \Pi) \) admits a NodeTrix planar representation with side assignment \( \Phi \) and permutation assignment \( \Pi \) is solvable in linear time [3].

In order to prove its NP-hardness, we reduce NAE3SAT to it. An instance of NAE3SAT consists of a collection of clauses on a set of Boolean variables, where each clause consists of exactly three literals. The problem asks whether there exists a truth assignment to the variables so that each clause has at least one true literal and at least one false literal. This problem has been shown to be NP-complete by Thomas J. Schaefer [11]. However, it is known to be polynomial when the graph of the adjacencies of the variables and clauses is planar [10].

Starting from a (non-planar) instance of NAE3SAT with variables \( x_1, x_2, \ldots, x_n \) and clauses \( C_1, C_2, \ldots, C_m \), we construct an instance \( G = (V, E, C, \Phi) \) of NodeTrix planarity testing with fixed sides as follows. First, we obtain a (non-planar) drawing \( \Gamma \) of the graph of its variables and clauses like the one in Fig. 3: The clause vertices are vertically aligned, the variable vertices are horizontally aligned, and the edges are drawn with L-shapes. Clearly this drawing can be computed in polynomial time and has a polynomial number of crossings.

![Fig. 3. A non-planar drawing of an instance of NAE3SAT. The small circled plus signs and circled minus signs represent direct and negated occurrences of the variables in the clauses, respectively.](image_url)

Then, we replace each vertex representing a variable with a variable gadget. The variable gadget for a variable of degree \( h \) is composed by \( h \) size-two clusters connected together as depicted in Fig. 4(a) and 4(b). Namely, let
\(V_{v,1}, V_{v,2}, \ldots, V_{v,h}\) be the \(h\) clusters of \(C\) composing the variable gadget of variable \(v\) and let \(\{u_{i,1}, u_{i,2}\}\) be the nodes of \(V_{v,i}\), with \(i = 1, \ldots, h\). We may encode a truth value with each one of the two possible representations of cluster \(V_i\); if in the matrix \(M_i\) representing \(V_i\) the column corresponding to the vertex \(u_{i,1}\) precedes the column corresponding to the vertex \(u_{i,2}\), we say that \(M_i\) is \textit{true}. Otherwise, we say that \(M_i\) is \textit{false}. Correspondingly, we say that \(\pi_i = u_{i,1}, u_{i,2}\) is the \textit{true} permutation of cluster \(V_i\) and that \(\pi_i = u_{i,2}, u_{i,1}\) is its \textit{false} permutation. In order to connect the clusters representing the variable gadget for \(v\), we add to \(E\), for \(i = 1, \ldots, h - 1\), the inter-cluster edges \(e_{i,1} = (u_{i,1}, u_{i+1,1})\) and \(e_{i,2} = (u_{i,2}, u_{i+1,2})\) and set \(\phi_i(e_{i,1}) = R\), \(\phi_i(e_{i,2}) = B\), \(\phi_{i+1}(e_{i,1}) = B\), \(\phi_{i+1}(e_{i,2}) = R\). It is immediate that in any NodeTrix planar representation of \(G\), all \(M_i\), \(i = 1, \ldots, h\), are either simultaneously \textit{true} or simultaneously \textit{false}. Correspondingly, we say that the variable gadget is \textit{true} or \textit{false}. Fig. 4(a) and 4(b) show an example of a \textit{true} and of a \textit{false} drawing of a variable gadget.

![Fig. 4.](image)

(a) The \textit{true} configuration of a variable gadget for a variable of degree four. (b) The \textit{false} configuration. (c) The \textit{not} gadget transforming an encoded \textit{true} value into a \textit{false} value. (d) The \textit{not} gadget transforming an encoded \textit{false} value into a \textit{true} value.

Each edge \(e\) attaching to a variable \(v\) in the drawing \(\Gamma\) (refer to Fig. 3) corresponds to two ‘parallel’ inter-cluster edges \(e_1\) and \(e_2\) attached to one of the clusters composing the variable gadget of \(v\). Let \(V_j\) be such a cluster. We set \(\phi_j(e_1) = T\) and \(\phi_j(e_2) = T\). Observe that the order in which \(e_1\) and \(e_2\) exit \(M_j\) depends on the truth value encoded by \(M_j\), and, hence, on the truth value encoded by the variable gadget for \(v\).

Fig. 3 depicts the gadget we use to replace crossings in \(\Gamma\), consisting of a cluster \(V_x\) of size three. From the figure it is apparent that, in any representation of \(V_x\), the left-to-right order of the edges entering \(M_x\) from the bottom side is the same as the left-to-right order of the edges exiting \(M_x\) from the top side. An analogous consideration holds for the top-to-bottom order of the edges entering \(M_x\) from the right side and the top-to-bottom order of the edges exiting \(M_x\) from the left side. This implies that the truth value encoded by the edges entering \(M_x\) is the same as the truth value encoded by the edges exiting it.

![Fig. 5.](image)

The six possible configurations of a crossing gadget.

We now describe the clause gadget. We assume that three pairs of edges, encoding the truth value of the variables occurring in the clause, arrive to the clause gadget. Let \(v_1, v_2,\) and \(v_3\) be the three variables whose literals \(l_1, l_2,\) and \(l_3\) occur in clause \(C\). Before entering the clause gadget, if literal \(l_i\) is a negated literal of variable \(v\) (i.e., if \(l_i = \overline{v}_i\)) then we attach the edges coming from \(v_i\) to a \textit{not} gadget, depicted in Fig. 4(c) and 4(d), and use the edges exiting the \textit{not} gadget instead of the edge coming directly from variable \(v_i\). This has the effect that all the three pairs of edges entering the clause gadget encode a truth value that is \textit{true} if the literal is \textit{true} and \textit{false} if the literal is \textit{false}. Correspondingly, the order in which \(v_1, v_2,\) and \(v_3\) occur in clause \(C\) is encoded by the permutation \(\sigma = \pi_{l_1} \pi_{l_2} \pi_{l_3}\), where \(\pi_{l_i}\) is the permutation for variable \(v_i\). 

![Fig. 6.](image)

The six possible configurations of a clause gadget.
false. In the following, therefore, we will consider the truth values of the literals, rather than the truth values of the variables.

The clause gadget, depicted in Fig. 6, is composed by three clusters $V_1, V_2$, and $V_3$ of size three, each having vertices $\{u_{i,a}, u_{i,b}, u_{i,c}\}$, $i = 1, 2, 3$. The three clusters are connected together in such a way that, in any NodeTrix planar representation of $G$, their permutations $\pi_i$, $i = 1, 2, 3$, always present the same sequence of the labels $a, b, c$. For example, if $\pi_1 = u_{1,c}, u_{1,a}, u_{1,b}$, then also $\pi_2 = u_{2,c}, u_{2,a}, u_{2,b}$ and $\pi_3 = u_{3,c}, u_{3,a}, u_{3,b}$.

For $i = 1, \ldots, 3$, the edges encoding the truth value of literal $l_i$ attach to cluster $V_i$, where the prescribed side is the right side of matrix $M_i$. The two edges $e_{1,1}$ and $e_{1,2}$ encoding the truth value of the literal $l_1$ attach to $u_{1,a}$ and $u_{1,b}$, respectively. The two edges $e_{2,1}$ and $e_{2,2}$ encoding the truth value of the literal $l_2$ attach to $u_{2,b}$ and $u_{2,c}$, respectively. Finally, the two edges $e_{3,1}$ and $e_{3,2}$ encoding the truth value of the literal $l_3$ attach to $u_{3,c}$ and $u_{3,a}$, respectively. Hence, if literal $l_1$ is true, matrix $M_1$ must have a permutation of its columns such that column $a$ precedes column $b$, while if literal $l_1$ is false, matrix $M_1$ must have a permutation of its columns such that column $a$ follows column $b$. Analogously, the truth value of literal $l_2$ determines whether in matrix $M_2$ column $b$ precedes or follows column $c$, and the truth value of literal $l_3$ determines whether in matrix $M_3$ column $c$ precedes or follows column $a$.

![Fig. 6. The clause gadget.](image)

It follows that, if all three literals are true, then they induce unsatisfiable constraints on the ordering of the columns of the matrices, since column $a$ should precede column $b$, $b$ should precede column $c$, and $c$ should precede $a$. The same holds if all three literals are false. It can be easily checked that, for any other combination of truth values of the literals, there exists an ordering of the columns of matrices $M_1$, $M_2$, and $M_3$ that makes a planar drawing of the edges possible. Therefore, the constructed instance of NodeTrix planarity with fixed sides admits a planar NodeTrix representation if and only if the original instance of NAE3SAT admits a solution.

A.6 Proof of Theorem 5

**Theorem 5.** NAE3SAT is NP-complete for triconnected Boolean formulas.

**Proof.** We use a reduction strategy that is similar to the one used in [8] to prove the hardness of PLANAR TRICONNECTED 3SAT. Here, however, we do not have to worry about planarity, since we are searching for a non-planar triconnected instance of NAE3SAT. Let $\varphi$ be an instance of NAE3SAT such that the graph $G$ that represents the occurrences of the variables into the clauses of $\varphi$ is not triconnected and, possibly, not biconnected and not connected. We show how to construct an instance $\varphi^*$ such that its graph $G^*$ is triconnected. The strategy consists of adding to $\varphi$ a suitable number of variables and clauses such that the obtained instance $\varphi^*$ is triconnected and has a solution if and only if $\varphi$ has one.

Consider an edge $e = (x, C)$ in the graph $G$ (refer to Fig. 7(a)). Suppose that variable $x$ occurs in clause $C$ with its direct literal (negated literal, respectively). We remove $e$ (that is, we remove the occurrence of $x$ into clause $C$) and replace it with the occurrence of the direct literal (negated literal, respectively) of a new variable $x'$. Also we add a new variable $y$ and the two clauses $(\overline{x} \lor x' \lor \overline{y})$ and $(\overline{x} \lor x' \lor \overline{y})$ (see Fig. 7(b)).

It can be shown that, in any assignment of truth values to the variables such that all clauses are satisfied in the NAE3SAT sense, the truth value of $x$ has to be the same as the truth value of $x'$. In fact, whatever is the truth
value of variable $y$, literals $\pi$ and $x'$ occur in a clause with a negative literal and a positive literal. Hence, they cannot be both positive or both negative. This implies $x = x'$. It follows that the obtained instance of NAE3SAT admits a solution if and only if the original instance admits one.

Now, consider two edges $e_1 = (x_1, C_1)$ and $e_2 = (x_2, C_2)$ of $G$ (see Fig. 7(c)). By replacing both edges with the above described gadget and by identifying the $y$ variables of the two gadgets, one obtains a bridge gadget for $e_1$ and $e_2$. The instance of NAE3SAT where $e_1$ and $e_2$ have been replaced by their bridge gadget is equivalent to the original instance and it has a path going from $C_1$ to $C_2$ and passing through $y$.

Hence, suppose you have a non-connected graph $G$ and let $e_1$ and $e_2$ be two edges belonging to two different connected components of $G$. Replacing $e_1$ and $e_2$ with their bridge gadget, the number of connected components of $G$ is decreased. Assume now that $G$ is connected. Let $T_{bcv}$ be the block-cut-vertex tree of $G$ and let $e_1$ and $e_2$ be two edges belonging to two different blocks of $T_{bcv}$. Replacing $e_1$ and $e_2$ with their bridge gadget, the number of blocks of $G$ is decreased (see for example Fig. 8). The same holds if we replace two edges $e_1$ and $e_2$ belonging to two triconnected components of $G$.

Therefore, we obtain a NAE3SAT instance $\varphi^*$ such that the graph that represents the occurrences of the variables into the clauses is triconnected and such that $\varphi^*$ admits a solution if and only if $\varphi$ admits one. Furthermore, the size of the gadgets used in the replacement is constant and therefore the size of $\varphi^*$ is polynomial in the size of $\varphi$. This concludes the proof.

\section{A.7 Proof of Theorem 6}

\textbf{Theorem 6.} NodeTrix planarity testing with free sides and cluster size at most $k$ is NP-complete for any $k \geq 5$.

\textbf{Proof.} The proof is based on a reduction from triconnected NAE3SAT. Starting from a triconnected instance $\varphi$ of NAE3SAT, we first build the instance $G_{fix}$ of NodeTrix planarity with fixed sides, analogously to the proof of
Theorem 4, but using the drawing strategy shown in Fig. 9 instead of that shown in Fig. 3. Observe that, since $\varphi$ is triconnected and because of the special planarization strategy we used, the frame graph of $G_{fix}$ is also triconnected. In fact, inserting crossing gadgets is equivalent to planarizing the drawing of Fig. 9.

![Fig. 9. A non-planar drawing of an instance of NAE3SAT such that, by replacing crossings with dummy nodes, you would obtain a planar and triconnected graph.](image)

Second, we obtain an instance $G_{free}$ of NodeTrix planarity with free sides by replacing each cluster of maximum size three, with the gadget depicted in Fig. 10(b) that uses exclusively clusters of size 5. Such a gadget consists of nine clusters such that the corresponding nodes in the frame graph of $G_{free}$ form a wheel graph with an external cycle of eight nodes all connected to a central one. Since the wheel has only one embedding (up to a flip), the gadget admits a NodeTrix planar representation only if the hub of the wheel is drawn inside the cycle formed by the other eight clusters. Also, the edges that in the instance $G_{fix}$ are constrained to attach to a specific side of a matrix due to the side assignment $\Phi$, are now all incident to the same cluster of the wheel. Since $G_{free}$ has a triconnected frame graph, the embedding of the frame graph is fixed, and a NodeTrix planar representation with fixed sides of $G_{fix}$ can be immediately obtained from a NodeTrix planar representation with free sides of $G_{free}$.

![Fig. 10. (a) A cluster $V_i$ of size three of the instance of NodeTrix planarity with fixed sides. (b) The gadget used to replace $V_i$ in the instance of NodeTrix planarity with free sides.](image)