NodeTrix Planarity Testing with Small Clusters

Emilio Di Giacomo[1(✉)], Giuseppe Liotta[1], Maurizio Patrignani[2], and Alessandra Tappini[1]

[1] Università degli Studi di Perugia, Perugia, Italy
{emilio.digiacomo,giuseppe.liotta}@unipg.it,
alessandra.tappini@studenti.unipg.it
[2] Roma Tre University, Rome, 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 $O(k^{3k+\frac{3}{2}}n^3)$ 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., [1–3,9]). 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.* [9] 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 subclusters) 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

F. Frati and K.-L. Ma (Eds.): GD 2017, LNCS 10692, pp. 479–491, 2018.
https://doi.org/10.1007/978-3-319-73915-1_37

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 $O(k^{3k+\frac{3}{2}} \cdot n^3)$-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.* [7] 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 Sect. 2. Sections 3 and 4 describe a polynomial time algorithm for clustered 2-trees with bounded cluster-size. In Sect. 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 Sect. 6. For reasons of space, some proofs are sketched or omitted and can be found in [6].

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., [4,8]).

A *flat clustered graph* $G = (V, E, \mathcal{C})$ is a simple graph with vertex set V, edge set E, and a partition $\mathcal{C}$ of V into sets $V_1, \ldots, V_h$, 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 (on M_j) belongs to the column or to the row of 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. Figure 1(a) is an example of a NodeTrix planar representation.

A formal definition of the problem investigated in the paper is as follows. Let $G = (V, E, \mathcal{C})$ be a flat clustered graph with n vertices and let k be the maximum cardinality of a cluster in $\mathcal{C}$. 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 $p_{v,\mathrm{T}}, p_{v,\mathrm{B}}, p_{v,\mathrm{L}}$, and $p_{v,\mathrm{R}}$ 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* $V_i \in \mathcal{C}$ 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 $\phi_i \colon \bigcup_{j \neq i} E_{i,j} \rightarrow \{\mathrm{T}, \mathrm{B}, \mathrm{L}, \mathrm{R}\}$, 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* $\mathcal{C}$ is a set Φ of side assignments for each $V_i \in \mathcal{C}$.

We denote as $G = (V, E, \mathcal{C}, \Phi)$ a flat clustered graph $G = (V, E, \mathcal{C})$ with a given side assignment $\Phi = \{\phi_1, \phi_2, \ldots, \phi_{|\mathcal{C}|}\}$. Let Γ 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 Γ are exactly the points $p_{u,\phi_i(e)}$ and $p_{v,\phi_j(e)}$, respectively. We call Γ a NodeTrix representation of G *consistent with* Φ. We say that $G = (V, E, \mathcal{C}, \Phi)$ is *NodeTrix planar* if it admits a NodeTrix planar representation consistent with Φ.

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 $G = (V, E, \mathcal{C})$ replaces e with a path $u_0 = u, u_1, u_2 = v$ and defines a new flat clustered graph $G' = (V', E', \mathcal{C}')$, where $V' = V \cup \{u_1\}$, $E' = E/e \cup \{(u_0, u_1), (u_1, u_2)\}$, and $\mathcal{C}' = \mathcal{C} \cup \{u_1\}$. 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 $G = (V, E, \mathcal{C})$ is the graph F obtained by collapsing each cluster $V_i \in \mathcal{C}$, 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 Sects. 4 and 5 are based on decomposing the planar frame F of a clustered graph $G = (V, E, \mathcal{C}, \Phi)$ 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 μ of the decomposition tree of a block of F we test whether the subgraph of G whose frame is the pertinent graph of μ satisfies the planar constraints imposed by the side assignment on the inter-cluster edges. A key ingredient to efficiently perform the test at μ is the notion of *wheel replacement.*

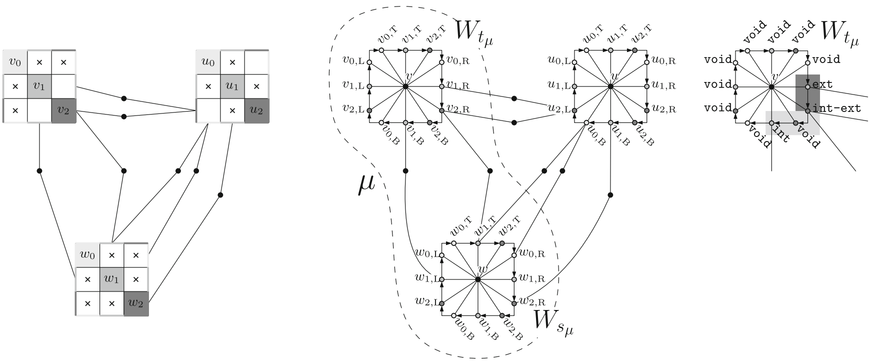
Let $G = (V, E, \mathcal{C}, \Phi)$ be a clustered graph with side assignment Φ and let $V_i \in \mathcal{C}$ be a cluster with $k > 1$ vertices. V_i admits $k!$ permutations of its vertices and we associate a suitable graph to each such permutation. Let $\pi_i = v_0, v_1, \ldots, v_{k-1}$ be a permutation of the vertices of V_i. The *wheel of V_i consistent with π_i* is the wheel graph consisting of a vertex v of degree $4k$ adjacent to the vertices of an oriented cycle $v_{0,\mathrm{T}}$, $v_{1,\mathrm{T}}$, $\ldots, v_{k-1,\mathrm{T}}$, $v_{0,\mathrm{R}}, v_{1,\mathrm{R}}, \ldots, v_{k-1,\mathrm{R}}$, $v_{k-1,\mathrm{B}}, v_{k-2,\mathrm{B}}, \ldots, v_{0,\mathrm{B}}$, $v_{k-1,\mathrm{L}}, v_{k-2,\mathrm{L}}, \ldots, v_{0,\mathrm{L}}$ 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 π_i. More precisely, a *wheel replacement of cluster V_i consistent with π_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 π_i; and (iii) for each inter-cluster edge $e = (u, v_j)$, with $v_j \in V_i$, insert edge $(u, v_{j,\phi_i(e)})$ incident to W_i. We call edge $(u, v_{j,\phi_i(e)})$ the *image* of edge $e = (u, v_j)$.

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

Theorem 1. *Let $G = (V, E, \mathcal{C}, \Phi, \Pi)$ be a clustered graph with side assignment Φ and permutation assignment Π. 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.*

Figure 1(a) and (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 $G = (V, E, \mathcal{C}, \Phi)$ for NodeTrix planarity by exploring the space of the possible permutation sets Π and corresponding wheel reductions in search of a NodeTrix planar $G = (V, E, \mathcal{C}, \Phi, \Pi)$. 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



(a) (b) (c)

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_{t_μ}; the complete internal and external sequences are highlighted.

described in [7], a brute-force approach that repeats this algorithm on each possible wheel reduction may lead to testing planarity on $|\mathcal{C}|^{k!}$ different instances. Instead, for each visited node μ of the decomposition tree T we compute a succinct description of the possible NodeTrix planar representations of the subgraph G_μ of G represented by the subtree of T rooted at μ. This is done by storing for the poles of μ those pairs of wheel graphs that are compatible with a NodeTrix planar representation of G_μ. 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 $G = (V, E, \mathcal{C}, \Phi)$ when the maximum size of any cluster of $\mathcal{C}$ 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.

Let $G = (V, E, \mathcal{C}, \Phi)$ be a series-parallel clustered graph with side assignment Φ 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 μ be a node of T, and let s_μ, t_μ be the poles of μ. Consider the pertinent graph F_μ represented by the subtree of T rooted at μ and let v_μ be a pole of μ ($v_\mu \in \{s_\mu, t_\mu\}$). Pole v_μ in the frame graph F may correspond to a non-trivial cluster V_i of $\mathcal{C}$. In this case, we call v_μ a *non-trivial pole of* μ and cluster V_i the *pertinent cluster* of v_μ.

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

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

A clockwise sequence $v_0, v_1, \ldots, v_j$ of vertices of the external cycle of W_μ is an *external sequence of pole* v_μ *consistent with* π_μ if v_0 and v_j are labeled either `ext` or `int-ext` and all the other vertices of the sequence are labeled either `void` or `ext`. An external clockwise sequence of pole v_μ is *complete* if it contains all the vertices of W_μ that are labeled `ext` and `int-ext`. Note that a complete external sequence may contain many `void` vertices but no `int` vertex. *Internal* and *complete internal* sequences of pole v_μ are defined analogously. Observe that a complete internal sequence and a complete external sequence of v_μ may not exist when vertices labeled `int` and vertices labeled `ext` alternate more than twice when traversing clockwise the external cycle of W_μ, or when three vertices are labeled `int-ext`. A special case is when W_μ has exactly two vertices w_1 and w_2 labeled `int-ext` and all other vertices are `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 $G = (V, E, \mathcal{C}, \Phi)$ for NodeTrix planarity, we implicitly take into account all possible permutation assignments Π by considering, for each non-trivial pole w_μ of each node μ 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 μ of T with information regarding the complete internal and complete external sequences of its non-trivial poles. Let μ be an internal node of T, let v_μ be a non-trivial pole of μ, let π_{v_μ} be a permutation of the pertinent cluster V_μ of v_μ, and let W_μ be the wheel of V_μ consistent with π_{v_μ}. We denote as $ISeq(\mu, v_\mu, \pi_{v_\mu})$ the complete internal sequence of v_μ consistent with π_{v_μ} in pole μ and as $ESeq(\mu, v_\mu, \pi_{v_\mu})$ the complete external sequence of v_μ consistent with π_{v_μ} in pole μ. We distinguish between the different types of nodes of T.

Node μ 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 μ. One end-vertex of e is the representative vertex in F of the pertinent cluster of the non-trivial pole v_μ. 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 ϕ_{v_μ} defines whether e is incident to the top, bottom, left, or right copy u_W of u in the wheel W_μ of V_μ. For any possible permutation π_{v_μ} we have $ISeq(\mu, v_\mu, \pi_{v_\mu}) = u_W$. If u_W is labeled `int-ext`, then $ESeq(\mu, v_\mu, \pi_{v_\mu})$ is the external cycle of W_μ starting at u_W and ending at u_W. Otherwise, traverse the external cycle of W_μ starting at u_W 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 μ is a P-node. Let $\nu_0, \nu_1, \dots, \nu_{h-1}$ be the children of μ. Observe that v_μ is a non-trivial pole also for the children $\nu_0, \nu_1, \dots, \nu_{h-1}$ of μ. We consider every permutation π_{v_μ} such that $\nu_0, \nu_1, \dots, \nu_{h-1}$ have both a complete internal sequence and a complete external sequence compatible with π_{v_μ}. The complete internal sequence of v_μ consistent with π_{v_μ} is the union of the complete internal sequences of the children $\nu_0, \nu_1, \dots, \nu_{h-1}$, that is $ISeq(\mu, v_\mu, \pi_{v_\mu}) = \cup_{i=0}^{h-1} ISeq(\nu_i, v_\mu, \pi_{v_\mu})$.

To determine the complete external sequence of v_μ consistent with π_{v_μ} we consider the intersection of the complete external sequences of the children of μ. 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_μ does not have a complete external sequence consistent with π_{v_μ}.

Node μ is an S-node. Let ν be the child of μ that shares the pole v_μ with μ. We consider every permutation π_{v_μ} such that ν has both $ISeq(\nu, v_\mu, \pi_{v_\mu})$ and $ESeq(\nu, v_\mu, \pi_{v_\mu})$. The complete internal (external) sequence of v_μ consistent with π_{v_μ} 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 μ with poles s_μ and t_μ, we check whether each possible pair $(\pi_{s_\mu}, \pi_{t_\mu})$ induces complete internal and external sequences for s_μ and t_μ 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 π_{s_μ} (π_{t_μ}, respectively) be a permutation such that s_μ (t_μ, respectively) has both a complete internal sequence and a complete external sequence compatible with π_{s_μ} (π_{t_μ}, respectively). We say that $(\pi_{s_\mu}, \pi_{t_\mu})$ is a

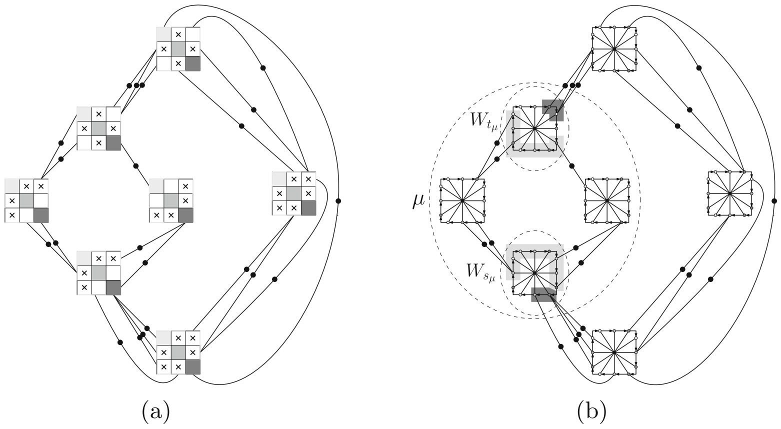


Fig. 2. (a) A NodeTrix planar representation Γ of $G = (V, E, \mathcal{C}, \Phi)$. (b) Γ induces a permutation assignment and a planar embedding of a wheel reduction of G; the complete internal and external sequences for a pair of poles are also highlighted.

compatible pair of permutations for μ if either one of the poles is a trivial pole or one of the following cases applies.

Node μ is a Q-node. In this case all $k!$ possible pairs of permutations for s_μ or t_μ (recall that only one of them is non-trivial) are compatible for μ.

Node μ is a P-node. Let $\nu_0, \nu_1, \dots, \nu_{h-1}$ be the children of μ. Consider a pair of permutations $(\pi_{s_\mu}, \pi_{t_\mu})$; we recall that, for $i = 0, \dots, h-1$, each ν_i has poles s_μ and t_μ. A first condition for pair $(\pi_{s_\mu}, \pi_{t_\mu})$ to be a compatible pair for μ is that $(\pi_{s_\mu}, \pi_{t_\mu})$ is also a compatible pair for ν_i, with $i = 0, \dots, h-1$. A second condition asks that the pair $(\pi_{s_\mu}, \pi_{t_\mu})$ *defines opposite orders on the poles of* μ. Namely, let W^s_μ (resp., W^t_μ) be the wheel of V_{s_μ} (resp., V_{t_μ}) consistent with π_{s_μ} (resp., π_{t_μ}). Traversing clockwise the external cycle of W^s_μ starting from the first vertex of $ESeq(\mu, s_\mu, \pi_{s_\mu})$, let $ISeq(\nu_0, s_\mu, \pi_{s_\mu})$, $ISeq(\nu_1, s_\mu, \pi_{s_\mu})$, $\dots$, $ISeq(\nu_{h-1}, s_\mu, \pi_{s_\mu})$ be the order by which the internal sequences are encountered. Pair $(\pi_{s_\mu}, \pi_{t_\mu})$ defines opposite orders on the poles of μ if, traversing clockwise the external cycle of W^t_μ starting from the first vertex of $ESeq(\mu, t_\mu, \pi_{s_\mu})$, the order by which we encounter the internal sequences of $\nu_0, \nu_1, \dots, \nu_{h-1}$ is the opposite one, i.e., the order is $ISeq(\nu_{h-1}, t_\mu, \pi_{t_\mu})$, $ISeq(\nu_{h-2}, t_\mu, \pi_{t_\mu})$, $\dots$, $ISeq(\nu_0, t_\mu, \pi_{t_\mu})$.

Node μ is an S-node. Let ν_0 and ν_1 be the children of μ such that $s_{\nu_0} = s_\mu$, $t_{\nu_0} = s_{\nu_1}$, and $t_{\nu_1} = t_\mu$. A pair $(\pi_{s_\mu}, \pi_{t_\mu})$ is a compatible pair for μ if there exists a permutation $\pi_{t_{\nu_0}}$ such that the pair $(\pi_{s_\mu}, \pi_{t_{\nu_0}})$ is compatible for ν_0 and the pair $(\pi_{t_{\nu_0}}, \pi_{t_\mu})$ is compatible for ν_1.

Figure 2 suggests that a NodeTrix planar representation of a clustered graph G defines a permutation assignment Π such that, for every node μ of T, pair $(\pi_{s_\mu}, \pi_{t_\mu})$ is a compatible pair for μ.

Lemma 1. *Let $G = (V, E, \mathcal{C}, \Phi)$ be a clustered graph with side assignment Φ 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 Π such that, for every node μ of T with poles s_μ and t_μ, we have that permutation $\pi_{s_\mu} \in \Pi$ and permutation $\pi_{t_\mu} \in \Pi$ form a compatible pair of permutations for μ.*

Lemma 2. *Let $G = (V, E, \mathcal{C}, \Phi)$ be a series-parallel clustered graph with side assignment Φ. Let k be the maximum size of any cluster in $\mathcal{C}$ and let n be the cardinality of V. There exists an $O(k^{3k+\frac{3}{2}} \cdot n^2)$-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. 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 Π such that $G = (V, E, \mathcal{C}, \Phi, \Pi)$ is NodeTrix planar. We first equip each non-trivial pole v_μ of every node μ of T with its possible complete internal and complete external sequences. The maximum number of complete internal sequences of v_μ is $k!$. The same is true for the complete external sequences. If each complete (internal or external) sequence of pole v_μ is encoded by means of its first and last vertex in the clockwise order around W_{v_μ}, 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 Π such that any node μ 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 μ. For each pair $(\pi_{s_\mu}, \pi_{t_\mu})$ of permutations of the poles of μ we equip μ with the information about whether such pair is compatible for μ. This requires $O(k!^2)$ space. If μ is a Q-node, every pair of permutations $(\pi_{s_\mu}, \pi_{t_\mu})$ is compatible for μ. It follows that all compatible pairs for μ can be computed in $O(k!)$ time (recall that one between s_μ and t_μ is non-trivial) and, hence, in $O(k! \cdot n)$ time for all the Q-nodes of T. If μ is a P-node with children $\nu_0, \nu_1, \ldots, \nu_{h-1}$, π_{s_μ} is one of the permutations that equip s_μ, and π_{t_μ} is one of the permutations that equip t_μ, testing whether the pair $(\pi_{s_\mu}, \pi_{t_\mu})$ is a compatible pair for μ can be executed in $O(h)$ time. It follows that all compatible pairs for μ 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 μ is an S-node with children ν_0 and ν_1, π_{s_μ} is one of the permutations that equip s_μ, and π_{t_μ} is one of the permutations that equip t_μ, testing whether the pair $(\pi_{s_\mu}, \pi_{t_\mu})$ is a compatible pair for μ can be executed in $O(k!)$ time, corresponding to choosing all possible permutations for the pole shared between ν_0 and ν_1. It follows that all compatible pairs for μ can be computed in $O(k!^3)$ time and, hence, in $O(k!^3 \cdot n)$ time for all S-nodes of T.

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 $O(k!^3 \cdot n^2)$. By Stirling's approximation, $k! \sim \sqrt{2\pi k}(\frac{k}{e})^k$ and thus a series-parallel clustered graph G with n vertices, side assignment $\varPhi$, and maximum cluster size k can be tested for NodeTrix planarity in $O(k^{3k+\frac{3}{2}} \cdot n^2)$ time. Note that the compatible pair of permutations stored at each node μ 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 $G = (V, E, \mathcal{C}, \varPhi)$ be a partial 2-tree clustered graph with side assignment $\varPhi$. Let k be the maximum size of any cluster in $\mathcal{C}$ 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 $\varPhi$ and if so, it computes a NodeTrix planar representation of G consistent with $\varPhi$.*

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 $\varPi$ such that, for every pair of poles s_μ and t_μ, $(\pi_{s_\mu}, \pi_{t_\mu})$ forms a compatible pair for μ when μ 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 $G = (V, E, \mathcal{C}, \varPhi)$ be a clustered graph with side assignment $\varPhi$, let F be the frame graph of G, and let T be the SPQR decomposition tree of F. Let μ be an R-node of T with poles s_μ and t_μ. A pair of permutations $(\pi_{s_\mu}, \pi_{t_\mu})$ forms a *compatible pair for* μ if there exists a planar embedding of the skeleton skel(μ) of μ for which the following conditions hold: (i) For each vertex v of skel(μ), let $e_0, e_1, \ldots e_{h-1}$ be the virtual edges of skel(μ) incident to v in clockwise order around v. Each such edge e_i is associated with a child ν_i of μ. There exists a permutation π_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 skel(μ) can be assigned a permutation π_v such that: $\pi_v = \pi_{s_\mu}$ if $v = s_\mu$ and $\pi_v = \pi_{s_\mu}$ if $v = t_\mu$, and for each virtual edge $e = (u, v)$ in skel(μ) that corresponds to a child ν of μ, the permutation pair (π_u, π_v) is compatible for ν.

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 skel(μ) are exactly two, denoted by π_v^+ and π_v^-. In order to test whether $(\pi_{s_\mu}, \pi_{t_\mu})$ forms a compatible pair for μ, we perform a traversal of skel(μ) starting at s_μ. Permutation π_{s_μ} and the clockwise order of the edges incident to s_μ can impose to choose only one of the two permutations π_w^+ o π_w^- available for each vertex w adjacent to s_μ 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_μ is reached. Therefore, testing whether π_{s_μ} and π_{t_μ} form a compatible pair for μ can be reduced to a suitable problem of labeling the edges and vertices of skel(μ) and verifying that at the end s_μ and t_μ are labeled with π_{t_μ} and π_{s_μ}.

Theorem 3. *Let $G = (V, E, \mathcal{C}, \Phi)$ be an n-vertex clustered graph with side assignment Φ such that the maximum size of any cluster in $\mathcal{C}$ 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.*

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. J. Graph Algorithms Appl. **21**(4), 731–755 (2017)
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: Hu, Y., Nöllenburg, M. (eds.) GD 2016. LNCS, vol. 9801, pp. 107–120. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50106-2_9
4. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing. Prentice Hall, Upper Saddle River (1999)
5. Di Giacomo, E., Liotta, G., Patrignani, M., Tappini, A.: NodeTrix Planarity Testing with Small Clusters. In: Frati, F., Ma, K. (eds.) GD 2017. LNCS, vol. 10692, pp. 479–491. Springer, Cham (2018)
6. Di Giacomo, E., Liotta, G., Patrignani, M., Tappini, A.: NodeTrix planarity testing with small clusters. CoRR 1708.09281 (2017). http://arxiv.org/abs/1708.09281
7. Gutwenger, C., Klein, K., Mutzel, P.: Planarity testing and optimal edge insertion with embedding constraints. J. Graph Algorithms Appl. **12**(1), 73–95 (2008)
8. Harary, F.: Graph Theory. Addison-Wesley Series in Mathematics. Addison Wesley, Reading (1969)
9. Henry, N., Fekete, J., McGuffin, M.J.: NodeTrix: a hybrid visualization of social networks. IEEE Trans. Vis. Comput. Graph. **13**(6), 1302–1309 (2007)