Computing k-Modal Embeddings of Planar Digraphs

Juan José Besa
University of California, Irvine, USA
`http://sites.uci.edu/juanbesa`
jjbesavi@uci.edu

Giordano Da Lozzo[1]
Roma Tre University, Rome, Italy
`http://www.dia.uniroma3.it/~dalozzo`
giordano.dalozzo@uniroma3.it

Michael T. Goodrich
University of California, Irvine, USA
`https://www.ics.uci.edu/~goodrich`
goodrich@uci.edu

Abstract

Given a planar digraph G and a positive even integer k, an embedding of G in the plane is *k-modal*, if every vertex of G is incident to at most k pairs of consecutive edges with opposite orientations, i.e., the incoming and the outgoing edges at each vertex are grouped by the embedding into at most k sets of consecutive edges with the same orientation. In this paper, we study the k-Modality problem, which asks for the existence of a k-modal embedding of a planar digraph. This combinatorial problem is at the very core of a variety of constrained embedding questions for planar digraphs and flat clustered networks.

First, since the 2-Modality problem can be easily solved in linear time, we consider the general k-Modality problem for any value of $k > 2$ and show that the problem is NP-complete for planar digraphs of maximum degree $\Delta \geq k+3$. We relate its computational complexity to that of two notions of planarity for flat clustered networks: Planar Intersection-Link and Planar NodeTrix representations. This allows us to answer in the strongest possible way an open question by Di Giacomo et al. [GD17], concerning the complexity of constructing planar NodeTrix representations of flat clustered networks with small clusters, and to address a research question by Angelini et al. [JGAA17], concerning intersection-link representations based on geometric objects that determine complex arrangements. On the positive side, we provide a simple FPT algorithm for partial 2-trees of arbitrary degree, whose running time is exponential in k and linear in the input size. Second, motivated by the recently-introduced planar L-drawings of planar digraphs [GD17], which require the computation of a 4-modal embedding, we focus our attention on $k = 4$. On the algorithmic side, we show a complexity dichotomy for the 4-Modality problem with respect to Δ, by providing a linear-time algorithm for planar digraphs with $\Delta \leq 6$. This algorithmic result is based on decomposing the input digraph into its blocks via BC-trees and each of these blocks into its triconnected components via SPQR-trees. In particular, we are able to show that the constraints imposed on the embedding by the rigid triconnected components can be tackled by means of a small set of reduction rules and discover that the algorithmic core of the problem lies in special instances of NAESAT, which we prove to be always NAE-satisfiable – a result of independent interest that improves on Porschen et al. [SAT03]. Finally, on the combinatorial side, we consider outerplanar digraphs and show that any such a digraph always admits a k-modal embedding with $k = 4$ and that this value of k is best possible for the digraphs in this family.

2012 ACM Subject Classification Human-centered computing → Graph drawings; Mathematics of computing → Graph algorithms; Information systems → Data structures

Keywords and phrases Modal Embeddings, Planarity, Directed Graphs, SPQR trees, NAESAT

[1] Corresponding author

27th Annual European Symposium on Algorithms (ESA 2019).
Editors: Michael A. Bender, Ola Svensson, and Grzegorz Herman; Article No. 19; pp. 19:1–19:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.19

Related Version Due to space limitations, some theorems and proofs are omitted or sketched and can be found in the full version of the paper [21], which is available at `https://arxiv.org/abs/1907.01630`.

Funding This work was supported in part by NSF grant 1815073, by MIUR Project "MODE" under PRIN 20157EFM5C, by MIUR Project "AHeAD" under PRIN 20174LF3T8, by MIUR-DAAD JMP N° 34120, and by H2020-MSCA-RISE project 734922 – "CONNECT".

1 Introduction

Computing k-modal embeddings of planar digraphs, for some positive even integer k called *modality*, is an important algorithmic task at the basis of several types of graph visualizations. In 2-*modal embeddings*, also called *bimodal embeddings*, the outgoing and the incoming edges at each vertex form two disjoint sequences. Bimodal embeddings are ubiquitous in Graph Drawing. For instance, level planar drawings [15, 22] and upward-planar drawings [8, 17] – two of the most deeply-studied graph drawing standards – determine bimodal embeddings. *4-modal embeddings*, where the outgoing and the incoming edges at each vertex form up to four disjoint sequences with alternating orientations, arise in the context of planar L-drawings of digraphs. In an *L-drawing* of an n-vertex digraph, introduced by Angelini et al. [1], vertices are placed on the $n \times n$ grid so that each vertex is assigned a unique x-coordinate and a unique y-coordinate and each edge uv (directed from u to v) is represented as a 1-bend orthogonal polyline composed of a vertical segment incident to u and of a horizontal segment incident to v. Recently, Chaplick et al. [13] addressed the question of deciding the existence of *planar L-drawings*, i.e., L-drawings whose edges might possibly overlap but do not cross and observe that the existence of a 4-modal embedding is a necessary condition for a digraph to admit such a representation (Fig. 1a).

To the best of our knowledge, no further relationships have been explicitly pointed out in the literature between modal embeddings and notable drawing models for modality values greater than four, yet they do exist. Da Lozzo et al. [14] and Di Giacomo et al. [18] study the planarity of *NodeTrix representations* of flat clustered networks, a hybrid representational model introduced by Henry, Fekete, and McGuffin [19], where clusters and intra-cluster edges are represented as adjacency-matrices, with rows and columns for the vertices of each cluster, and inter-cluster edges are Jordan arcs connecting different matrices (Fig. 1b). For clusters containing only two vertices, it is possible to show that the problem of computing planar NodeTrix representations coincides with the one of testing whether a special digraph, called the *canonical digraph*, associated to the network admits a 6-modal embedding. For higher values of modality, k-modal embeddings occur in the context of Intersection-Link representations of flat clustered networks. In an *intersection-link representation* [3, 5], vertices are represented as translates of the same polygon, intra-cluster edges are represented

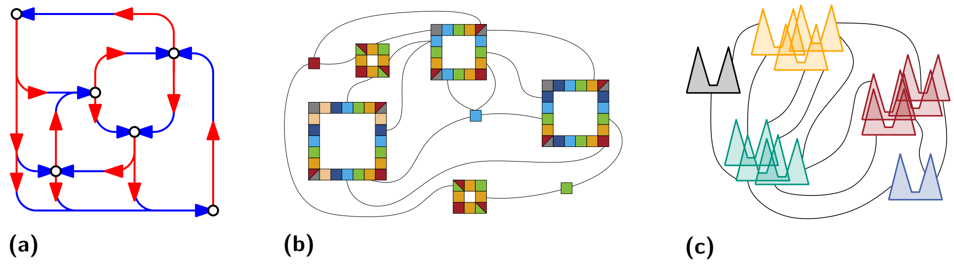

Figure 1 (a) A planar L-drawing, which determines a 4-modal embedding. (b) A planar NodeTrix representation. (c) A planar intersection-link representation using comb-shaped polygons.

via intersections between the polygons corresponding to their endpoints, and inter-cluster edges – similarly to NodeTrix representations – are Jordan arcs connecting the polygons corresponding to their endpoints. For any modality $k \geq 2$, it can be shown that testing the existence of a k-modal embedding of the canonical digraph of a flat clustered network with clusters of size two is equivalent to testing the existence of an intersection-link representation in which the curves representing inter-cluster edges do not intersect, when vertices are drawn as comb-shaped polygons (Fig. 1c).

Related Work. It is common knowledge that the existence of bimodal embeddings can be tested in linear time: Split each vertex v that has both incoming and outgoing edges into two vertices v_{in} and v_{out}, assign the incoming edges to v_{in} and the outgoing edges to v_{out}, connect v_{in} and v_{out} with an edge, and test the resulting (undirected) graph for planarity using any of the linear-time planarity-testing algorithms [11, 20]. Despite this, most of the planarity variants requiring bimodality are NP-complete; for instance, upward planarity [17], windrose planarity [6], partial-level planarity [12], clustered-level planarity and T-level planarity [4, 23], ordered-level planarity and bi-monotonicity [23]. In this scenario, a notable exception is represented by the classic level planarity problem, which can be solved in linear time [22], and its generalizations on the standing cylinder [7], rolling cylinder and the torus [2]. Although the existence of a bimodal embedding is easy to test, Binucci, Didimo, and Giordano [9] prove that the related problem of finding the maximum bimodal subgraph of an embedded planar digraph is an NP-hard problem. Moreover, Binucci, Didimo, and Patrignani [10] show that, given a *mixed planar graph*, i.e., a planar graph whose edge set is partitioned into a set of directed edges and a set of undirected edges, orienting the undirected edges in such a way that the whole graph admits a bimodal embedding is an NP-complete problem. On the other hand, the question regarding the computational complexity of constructing k-modal embedding for $k \geq 4$ has not been addressed, although the related problem of testing the existence of planar L-drawings has been recently proved NP-complete [13].

Our results. We study the complexity of the k-Modality problem, which asks for the existence of k-modal embeddings of planar digraphs – with an emphasis on $k = 4$. Our results are as follows:

- We demonstrate a complexity dichotomy for the 4-Modality problem with respect to the maximum degree Δ of the input digraph. Namely, we show NP-completeness when $\Delta \geq 7$ (see [21, Section 9]) and give a linear-time testing algorithm for $\Delta \leq 6$ (Theorem 19). Further, we extend the hardness result to any modality value larger than or equal to 4, by proving that the k-Modality problem is NP-complete for $k \geq 4$ when $\Delta \geq k + 3$.
- We provide an FPT-algorithm for k-Modality that runs in $f(k)O(n)$ time for the class of directed partial 2-trees (Theorem 16), which includes series-parallel and outerplanar digraphs.
- In Section 3, we relate k-modal embeddings with hybrid representations of flat clustered graphs, and exploit this connection to give new complexity results (Theorems 4 and 8) and algorithms (Theorems 3 and 7) for these types of representations. In particular, our NP-hardness results allow us to answer two open questions. Namely, we settle in the strongest possible way an open question, posed by Di Giacomo et al. [18, Open Problem (i)], about the complexity of computing planar NodeTrix representations of flat clustered graphs with clusters of size smaller than 5. Also, we address a research question by Angelini et al. [3, Open Problem (2)] about the representational power of intersection-link representations based on geometric objects that give rise to complex combinatorial structures, and solve it when the considered geometric objects are k-combs.
- Finally, in [21, Section 10], we show that not every outerplanar digraph admits a bimodal embedding, whereas any outerplanar (multi-)digraph admits a 4-modal embedding.

The algorithms presented in this paper employ the SPQ- and SPQR-tree data structures to succinctly represent the exponentially-many embeddings of series-parallel and biconnected planar digraphs, respectively, and can be easily modified to output an embedding of the input digraph in the same time bound. In particular, our positive result for $\Delta \leq 6$ is based on a set of simple reduction rules that exploit the structure of the rigid components of bounded-degree planar digraphs. These rules allow us to tackle the algorithmic core of the problem, by enabling a final reduction step to special instances of NAESAT, previously studied by Porschen et al. [26], which we prove to be always NAE-satisfiable [21, Section 7].[2]

2 Definitions

We assume familiarity with basic concepts concerning directed graphs, planar embeddings, connectivity and the BC-tree data structure; see the full version [21] for more details.

Directed graphs. A *directed graph* (for short *digraph*) $G = (V, E)$ is a pair, where V is the set of vertices and E is the set of *(directed) edges* of G, i.e., ordered pairs of vertices. We also denote the sets V and E by $V(G)$ and $E(G)$, respectively. The *underlying graph* of G is the undirected graph obtained from G by disregarding edge directions. Let v be a vertex, we denote by $E(v)$ the set of edges of G incident to v and by $\deg(v) = |E(v)|$ the *degree* of v. For an edge $e = uv$ directed from u to v and an end-point $x \in \{u, v\}$ of e, we define the *orientation* $\sigma(e, x)$ of e *at* x as $\sigma(e, x) =$ ⫰, if $x = u$, and $\sigma(e, x) =$ ⫯, if $x = v$, and we say that uv is *outgoing from* u and *incoming at* v.

Modality. Let G be a planar digraph and let $\mathcal{E}$ be an embedding of G. A pair of edges e_1, e_2 that appear consecutively in the circular order around a vertex v of G is *alternating* if they do not have the same orientation at v, i.e., they are not both incoming at or both outgoing from v. Also, we say that vertex v is k*-modal*, or that v has *modality* k, or that the *modality of* v *is* k in $\mathcal{E}$, if there exist exactly k alternating pairs of edges incident to v in $\mathcal{E}$. Clearly, the value k needs to be a non-negative even integer. An embedding of a digraph G is k*-modal*, if each vertex is at most k-modal; see Fig. 3(left).

We now define an auxiliary problem, called k-MaxModality (where k is a positive even integer), which will be useful to prove our algorithmic results. We denote the set of non-negative integers by $\mathbb{Z}^*$ and the set of non-negative even integers smaller than or equal to k as $\mathbb{E}^+_k = \{b : b = 2a, b \leq k, a \in \mathbb{Z}^*\}$. Given a graph G, we call *maximum-modality function* an integer-valued function $m : V(G) \rightarrow \mathbb{E}^+_k$. We say that an embedding $\mathcal{E}$ of G *satisfies* m *at a vertex* v if the modality of v in $\mathcal{E}$ is at most $m(v)$.

Problem: k-MaxModality

Input: A pair $\langle G, m \rangle$, where G is a digraph and m is a maximum-modality function.
Question: Is there an embedding $\mathcal{E}$ of G that *satisfies* m at every vertex?

[2] In "*Stefan Porschen, Bert Randerath, Ewald Speckenmeyer: Linear Time Algorithms for Some Not-All-Equal Satisfiability Problems. SAT 2003: 172-187*" [26], the authors state in the abstract "First we show that a NAESAT model (if existing) can be computed in linear time for formulas in which each variable occurs at most twice.". We give a strengthening of this result by showing that the only negative formulas with the above properties are those whose variable-clause graph contains components isomorphic to a simple cycle and provide a recursive linear-time algorithm for computing a NAE-truth assignment for formulas in which each variable occurs at most twice, when one exists, which is also considerably simpler than the one presented in [26].

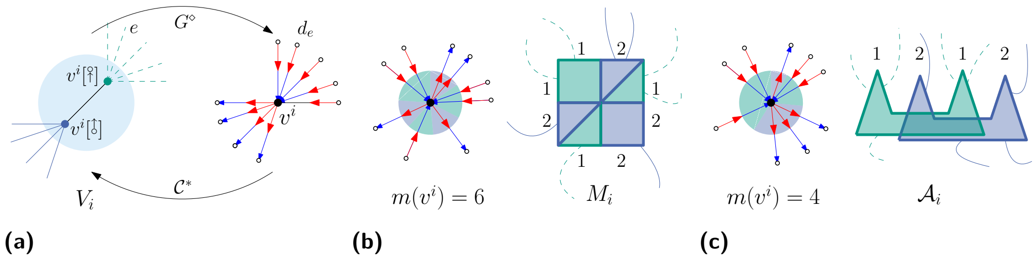

Figure 2 (a) Illustrations for the duality between the canonical digraph and the canonical c-graph. Correspondence (b) between 6-modal embeddings and planar NodeTrix representations, and (c) between 4-modal embeddings and clique-planar representations using 2-combs as geometric objects.

3 Implications on Hybrid Representations

A *flat clustered graph* (for short, *c-graph*) is a pair $\mathcal{C} = (G = (V, E), \mathcal{P} = (V_1, V_2, \dots, V_c))$, where G is a graph and $\mathcal{P}$ is a partition of V into sets V_i, for $i = 1, \dots, c$, 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 is an *inter-cluster edge*, if $i \neq j$. The problem of visualizing such graphs so to effectively convey both the relation information encoded in the set E of edges of G and the hierarchical information given by the partition $\mathcal{P}$ of the clusters has attracted considerable research attention. In a *hybrid representation* of a graph different conventions are used to represent the dense and the sparse portions of the graph [3, 5, 14, 18, 19, 24, 27]. We present important implications of our results on some well-known models for hybrid-representations of c-graphs.

Let $\mathcal{C}$ be a c-graph whose every cluster forms a clique of size at most 2, that is, each cluster contains at most two vertices connected by an intra-cluster edge. Starting from $\mathcal{C}$ we define an auxiliary digraph $G^\diamond$, called the *canonical digraph* for $\mathcal{C}$, as follows. Without loss of generality, assume that, for $i = 1, \dots, c$, each cluster V_i contains two vertices denoted as $v^i[⫯]$ and $v^i[⫰]$. The vertex set of $G^\diamond$ contains a vertex v^i, for $i = 1, 2, \dots, c$, and a dummy vertex d_e, for each inter-cluster edge $e \in E$. The edge set of $G^\diamond$ contains two directed edges, for each inter-cluster edge $e = (v^i_x, v^j_y) \in E$, with $x, y \in \{⫯, ⫰\}$ and $i \neq j$; namely, $E(G^\diamond)$ contains (i) either the directed edges $v^i_x d_e$, if $x = ⫰$, or the directed edge $d_e v^i_x$, if $x = ⫯$, and (ii) either the directed edges $v^i_y d_e$, if $y = ⫰$, or the directed edge $d_e v^i_y$, if $y = ⫯$.

Let now $D = (V, E)$ be a digraph. We construct a c-graph $\mathcal{C}^* = (G^* = (V^*, E^*), \mathcal{P}^*)$ from D whose every cluster forms a clique of size at most 2, called the *canonical c-graph* for D, as follows. For each vertex $v^i \in V$, G^* contains two vertices $v^i[⫯]$ and $v^i[⫰]$, which form the cluster $V_i = \{v[⫯], v[⫰]\}$ in $\mathcal{P}^*$. For each (directed) edge $v^i v^j$ of D, G^* contains an (undirected) edge $(v^i[⫰], v^j[⫯])$; that is, each directed edge in E that is incoming (outgoing) at a vertex v^i and outgoing (incoming) at a vertex v^j corresponds to an inter-cluster edge in E^* incident to $v^i[⫯]$ (to $v^i[⫰]$) and to $v^j[⫰]$ (to $v^j[⫯]$). Finally, for each vertex $v^i \in V$, G^* contains an intra-cluster edge $(v^i[⫯], v^i[⫰])$. The canonical digraph and the canonical c-graph form dual concepts, as illustrated in Fig. 2a; the canonical c-graph of $G^\diamond$ is the original c-graph $\mathcal{C}$ (neglecting clusters originated by dummy vertices) and the canonical digraph of $\mathcal{C}^*$ is the original digraph D (suppressing dummy vertices).

NodeTrix Planarity. A *NodeTrix representation* of a c-graph $\mathcal{C} = (G, \mathcal{P})$ is a drawing of $\mathcal{C}$ such that: **(i)** each cluster $V_i \in \mathcal{P}$ is represented as a symmetric adjacency matrix M_i (with $|V_i|$ rows and columns), drawn in the plane so that its boundary is a square Q_i with sides parallel to the coordinate axes; **(ii)** no two matrices intersect, that is, $Q_i \cap Q_j = \emptyset$, for all $1 \leq i < j \leq c$; **(iii)** each intra-cluster edge is represented by the adjacency matrix M_i; and **(iv)** each inter-cluster edge (u, v) with $u \in V_i$ and $v \in V_j$ is represented as a simple Jordan arc connecting a point on the boundary of Q_i with a point on the boundary of Q_j, where the point on Q_i (on Q_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 is *planar* if no inter-cluster edge intersects a matrix or another inter-cluster edge, except possibly at a common end-point; see Figs. 1b and 2b. The NodeTrix Planarity problem asks whether a c-graph admits a planar NodeTrix representation. NodeTrix Planarity has been proved NP-complete for c-graphs whose clusters have size larger than or equal to 5 [18].

We are ready to establish our main technical lemmas.

▶ **Lemma 1.** *C-graph $\mathcal{C}$ is NodeTrix planar if and only if $G^\diamond$ admits a 6-modal embedding.*

▶ **Lemma 2.** *Digraph D admits a 6-modal embedding if and only if $\mathcal{C}^*$ is NodeTrix planar.*

Proof sketch for Lemmas 1 and 2. Let M_i be the matrix representing cluster $V_i = \{v^i[\text{⫯}], v^i[\text{⫰}]\}$. We have that, independently of which of the two possible permutations for the rows and columns of M_i is selected, the boundary of Q_i is partitioned into three maximal portions associated with $v^i[\text{⫯}]$ and three maximal portions associated with $v^i[\text{⫰}]$; that is, they form the pattern $[1, 2, 1, 2, 1, 2]$, see Fig. 2b. Therefore, any planar NodeTrix representation of $\mathcal{C}$ (of $\mathcal{C}^*$) can be turned into a 6-modal embedding of $G^\diamond$ (of D) via a local redrawing procedure which operates in the interior of Q_i; also, any 6-modal embedding of $G^\diamond$ (of D) can be turned into a planar NodeTrix representation of $\mathcal{C}$ (of $\mathcal{C}^*$) via a local redrawing procedure which operates in a small disk centered at v_i that contains only v_i and intersects only edges incident to v_i.

Since $G^\diamond$ can be constructed in linear time from $\mathcal{C}$, Lemma 1 and the algorithm of Theorem 16 for solving k-Modality of directed partial 2-trees give us the following.

▶ **Theorem 3.** NodeTrix Planarity *can be solved in linear time for flat clustered graphs whose clusters have size at most* 2 *and whose canonical digraph is a directed partial* 2*-tree.*

Note that (i) $\mathcal{C}^*$ can be constructed in polynomial time from D, (ii) $\mathcal{C}^*$ only contains clusters of size 2 (although clusters corresponding to vertices of D incident to incoming or outgoing edges only could be simplified into clusters of size 1), and (iii) each cluster $V_i \in \mathcal{P}^*$, with $v^i \in V(D)$, is incident to α inter clusters edges, where α is the degree of v^i in D. These properties and the fact that in [21, Section 9] we prove the k-Modality problem to be NP-complete for digraphs of maximum degree $\Delta \geq k + 3$ give us the following.

▶ **Theorem 4.** NodeTrix Planarity *is NP-complete for flat clustered graphs whose clusters have size at most* 2*, even if each cluster is incident to at most* 9 *inter-cluster edges.*

We remark that the above NP-completeness result is best possible in terms of the size of clusters, as clusters of size 1 do not offer any advantage to avoid intersections between inter-cluster edges. Also, it solves [18, Open Problem (i)], which asks for the complexity of NodeTrix Planarity for c-graphs whose clusters have size between 2 and 5.

Clique Planarity. Hybrid representations have also been recently studied in the setting in which clusters are represented via intersections of geometric objects. In particular, Angelini et al. [3] introduced the following type of representations. Suppose that a c-graph $(G, \mathcal{P})$ is given, where $\mathcal{P}$ is *a set of cliques* that partition the vertex set of G. In an *intersection-link representation*, the vertices of G are represented by geometric objects that are translates of the same rectangle. Consider an edge (u, v) and let $R(u)$ and $R(v)$ be the rectangles representing u and v, respectively. If (u, v) is an intra-cluster edge (called *intersection-edge* in [3]), we represent it by drawing $R(u)$ and $R(v)$ so that they intersect, otherwise if (u, v) is an intra-cluster edge (called *link-edge* in [3]), we represent it by a Jordan arc connecting $R(u)$ and $R(v)$. A *clique-planar* representation is an intersection-link representation in which no inter-cluster edge intersects the interior of any rectangle or another inter-cluster edge, except possibly at a common end-point. The Clique Planarity problem asks whether a c-graph $(G, \mathcal{P})$ admits a clique-planar representation.

Angelini et al. proved the Clique Planarity problem to be NP-complete, when $\mathcal{P}$ contains a cluster V^* with $|V^*| \in O(|G|)$, and asked, in [3, Open Problem (2)], about the implications of using different geometric objects for representing vertices, rather than translates of the same rectangle. We address this question by considering k-combs as geometric objects, where a *k-comb* is the simple polygon with k spikes illustrated in Fig. 2c. We have the following.

▶ **Lemma 5.** *C-graph $\mathcal{C}$ is a positive instance of* Clique Planarity *using k-combs as geometric objects if and only if $G^\diamond$ admits a $2k$-modal embedding.*

▶ **Lemma 6.** *Digraph D admits an 4-modal embedding if and only if $\mathcal{C}^*$ is a positive instance of* Clique Planarity *using 2-combs as geometric objects.*

Proof sketch for Lemmas 5 and 6. Let A_i be an arrangements of 2-combs representing cluster $V_i = \{v^i[⫯], v^i[⫰]\}$. We have that, the boundary of A_i is partitioned into at most two maximal portions associated with $v^i[⫯]$ and at most two maximal portions associated with $v^i[⫰]$; that is, they form the pattern $[1, 2, 1, 2]$, see Fig. 2c. Therefore, as for Lemmas 1 and 2, we can exploit a local redrawing procedure to transform a clique-planar representation of $\mathcal{C}$ (of $\mathcal{C}^*$) into a 4-modal embedding of $G^\diamond$ (of D), and vice versa.

Combining Lemma 5 and the algorithm of Theorem 16 gives us the following positive result.

▶ **Theorem 7.** Clique Planarity *using r-combs, with $r \geq 1$, as geometric objects can be solved in linear time for flat clustered graphs whose clusters have size at most 2 and whose canonical digraph is a directed partial 2-tree.*

Finally, Lemma 6 and the discussion preceding Theorem 4 imply the following.

▶ **Theorem 8.** Clique Planarity *using 2-combs as geometric objects is NP-complete, even for flat clustered graphs with clusters of size at most 2 each incident to at most 7 inter-cluster edges.*

4 Polynomial-time Algorithms

In this section, we present an algorithmic framework to devise efficient algorithms for the k-Modality problem for notable families of instances. First, in Section 4.1, we show how to efficiently reduce the k-Modality problem in simply-connected digraphs to the k-MaxModality problem in biconnected digraphs. Then, in Section 4.2, we introduce preliminaries and definitions concerning SPQR-trees and k-modal embeddings of biconnected digraphs.

4.1 Simply-Connected Graphs

We first observe that the k-MaxModality problem is a generalization of the k-Modality problem. In fact, a directed graph $G = (V, E)$ admits a k-modal embedding if and only if the pair $\langle G, m \rangle$, with $m(v) = k, \forall v \in V(G)$, is a positive instance of the k-MaxModality problem.

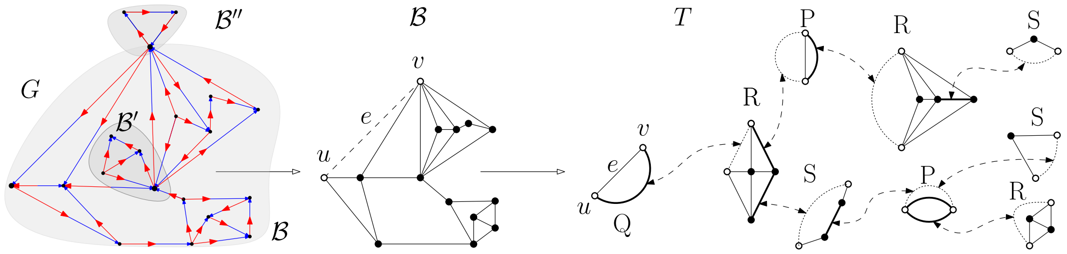

Figure 3 (left) A 4-modal embedding of a simply-connected planar digraph G. (right) The SPQR $\mathcal{T}$ of the block $\mathcal{B}$ of G rooted at the edge $e = uv$. The extended skeletons of all non-leaf nodes of $\mathcal{T}$ are shown; virtual edges corresponding to S-, P-, and R-nodes are thick.

▶ **Observation 1.** k-Modality *reduces in linear time to* k-MaxModality.

Let $\langle G, m : V(G) \to \mathbb{E}_4^+ \rangle$ be an instance of 4-MaxModality; also, let β be a leaf-block of the BC-tree $\mathcal{T}$ of G and let v be the parent cut-vertex of β in $\mathcal{T}$. We denote by G_β^- the subgraph of G induced by v and the vertices of G not in β, i.e., $G_\beta^- = G - (\beta - \{v\})$. Also, let $B(\mathcal{T})$ be the set of blocks in $\mathcal{T}$. We show that k-MaxModality (and k-Modality, by Observation 1) in simply-connected digraphs is Turing reducible to k-MaxModality in biconnected digraphs.

▶ **Theorem 9.** *Given a subroutine* TestBiconnected *that tests* k-MaxModality *for biconnected instances, there exists a procedure* TestSimplyConnected *that tests* k-MaxModality *for simply-connected digraphs. Further, given an instance* $\langle G, m \rangle$ *of* k-MaxModality*, the runtime of* TestSimplyConnected$(\langle G, m \rangle)$ *is*

$$O\big(|G| + \log k \sum_{\beta \in B(\mathcal{T})} r(\beta)\big),$$

where $r(\beta)$ *is the runtime of* TestBiconnected$(\langle \beta, m \rangle)$ *and* $\mathcal{T}$ *is the BC-tree of* G.

Sketch. The algorithm selects a leaf-block β of $\mathcal{T}$, with parent cut-vertex v, and finds an embedding of β with the minimum modality at v satisfying m at all vertices, by performing a binary search using the TestBiconnected procedure. We then remove β, replace G with G_β^-, and update the value of $m(v)$, so to account for the alternations at v introduced by β. The procedure terminates when all the blocks have been processed. Therefore, since the total number of calls to subroutine TestBiconnected is bounded by the number of blocks of G, which is $O(|\mathcal{T}|) = O(|G|)$ multiplied by $\log k$, the overall running time is $O(|G| + \log k \sum_{\beta \in B(\mathcal{T})} r(\beta))$. ◀

4.2 Biconnected Graphs

To handle the decomposition of a biconnected digraph into its triconnected components, we use SPQR-trees, a data structure introduced by Di Battista and Tamassia [16].

SPQR-trees. Let G be a biconnected digraph. We consider SPQR-trees that are rooted at an edge e of G, called the *reference edge*. The rooted SPQR-tree $\mathcal{T}$ of G with respect to e describes a recursive decomposition of G induced by its split pairs. The nodes of $\mathcal{T}$ are of four types: S, P, Q, and R. Each node μ of $\mathcal{T}$ has an associated undirected multigraph skel(μ), called the *skeleton* of μ, with two special nodes u_μ and v_μ (the *poles* of μ), and an associated subgraph pert(μ) of G, called *pertinent* of μ. The skeleton graph equipped with the edge $u_\mu v_\mu$, called the *parent edge*, is the *extended skeleton* of μ. Refer to Fig. 3(right).

Each edge of skel(μ), called *virtual edge*, is associated with a child of μ in $\mathcal{T}$. The skeleton of μ describes how the pertinent graphs of the children of μ have to be "merged" via their poles to obtain pert(μ). The extended skeleton of an S-, P-, R-, and Q-node is a cycle, parallel, triconnected graph, and a 2-gon, respectively. It follows that skeleton and pertinent graphs are always biconnected once the parent edge is added. A *series-parallel digraph* is a biconnected planar digraph whose SPQR-tree only contains S-, P-, and Q-nodes. A *partial 2-tree* is a digraph whose every block is a series-parallel digraph.

A digraph G is planar if and only if the skeleton of each R-node in the SPQR-tree of G is planar. By selecting *regular embeddings* for the skeletons of the nodes of $\mathcal{T}$, that is, embeddings in which the parent edge is incident to the outer face, we can construct any embedding of G with the edge e on the outer face, where the choices for the embeddings of the skeletons are all and only the (i) flips of the R-nodes and the (ii) permutations for the virtual edges of the P-nodes.

Consider a pair $\langle G, m\rangle$ such that G is biconnected and let $\mathcal{E}$ be a planar embedding of G. Also, let $\mathcal{T}$ be the SPQR-tree of G rooted at an edge e of G incident to the outer face of $\mathcal{E}$. We will assume that the virtual edges of the skeletons of the nodes in $\mathcal{T}$ are oriented so that the extended skeleton of each node μ is a DAG with a single source u_μ and a single sink v_μ. Let μ be a node of $\mathcal{T}$ and let $\mathcal{E}_\mu$ be the planar (regular) embedding of skel(μ) induced by $\mathcal{E}$. For an oriented edge $d = uv$ of skel(μ), the *left* and *right face* of d in $\mathcal{E}_\mu$ is the face of $\mathcal{E}_\mu$ seen to the left and to the right of d, respectively, when traversing this edges from u to v. We define the *outer left (right) face* of $\mathcal{E}_\mu$ as the left (right) face of the edge $u_\mu v_\mu$ in $\mathcal{E}_\mu$.

Embedding tuples. An *embedding tuple* (for short, *tuple*) is a 4-tuple $\langle \sigma_1, a, \sigma_2, b\rangle$, where $\sigma_1, \sigma_2 \in \{⫰, ⫯\}$ are orientations and $a, b \in \mathbb{N}$ are non-negative integers. Consider two tuples $t = \langle \sigma_1, a, \sigma_2, b\rangle$ and $t' = \langle \sigma'_1, a', \sigma'_2, b'\rangle$. We say that t *dominates* t', denoted as $t \preceq t'$, if $\sigma_1 = \sigma'_1$, $\sigma_2 = \sigma'_2$, $a \leq a'$, and $b \leq b'$. Also, we say that t and t' are *incompatible*, if none of them dominates the other. Since the relationship $\preceq$ is reflexive, antisymmetric, and transitive, it defines a poset $(T, \preceq)$, where T is the set of embedding tuples. A subset $S \subseteq T$ is *succinct* or an *antichain*, if the tuples in S are pair-wise incompatible. Consider two subsets $S, S' \subseteq T$ of tuples. We say that S *dominates* S', denoted as $S \preceq S'$, if for any tuples $t' \in S'$ there exists at least one tuple $t \in S$ such that $t \preceq t'$. Also, S *reduces* S' if $S \preceq S'$ and $S \subseteq S'$. Finally, S is a *gist* of S', if S is succinct and reduces S'.

Let e_u and e_v be the edges of pert(μ) incident to the outer left face of $\mathcal{E}_\mu$ and to u_μ and v_μ, respectively, possibly $e_u = e_v$. Also, let a and b be non-negative integers. We say that the embedding $\mathcal{E}_\mu$ *realizes* tuple $\langle \sigma_1, a, \sigma_2, b\rangle$, if $\sigma_1 = \sigma(e_u, u_\mu)$, $\sigma_2 = \sigma(e_v, v_\mu)$, and a and b are the number of inner faces of $\mathcal{E}_\mu$ whose (two) edges incident to u_μ and to v_μ, respectively, form an alternating pair. A tuple $t = \langle \sigma_1, a, \sigma_2, b\rangle$ is *realizable by* μ, if there exists an embedding of pert(μ) that realizes t, and *admissible*, if $a \leq m(u)$ and $b \leq m(v)$. A tuple is *good for* μ if it is both admissible and realizable by μ. We denote by $S(\mu)$ the gist of the set of good tuples for a node μ. Let e_μ be the virtual edge representing μ in the skeleton of the parent of μ in $\mathcal{T}$, with a small overload of notation, we also denote $S(\mu)$ by $S(e_\mu)$. For a tuple $t = \langle \sigma_1, a, \sigma_2, b\rangle \in S(e_\mu)$, where $e = u_\mu v_\mu$, the pair (σ_1, a) is the *embedding pair* of t at u_μ; likewise, the pair (σ_2, b) is the *embedding pair* of t at v_μ. We have the following substitution lemma.

▶ **Lemma 10.** *Let $\mathcal{E}$ be a planar embedding of G satisfying m. Let μ be a node of $\mathcal{T}$ and let $\mathcal{E}_\mu$ be the embedding of* pert(μ) *induced by $\mathcal{E}$. Also, let $\mathcal{E}'_\mu \neq \mathcal{E}_\mu$ be an embedding of* pert(μ) *satisfying m. Then, G admits an embedding $\mathcal{E}'$ satisfying m in which the embedding of* pert(μ) *is $\mathcal{E}'_\mu$, if $t' \preceq t$, where t and t' are the embedding tuples realized by $\mathcal{E}_\mu$ and by $\mathcal{E}'_\mu$, respectively.*

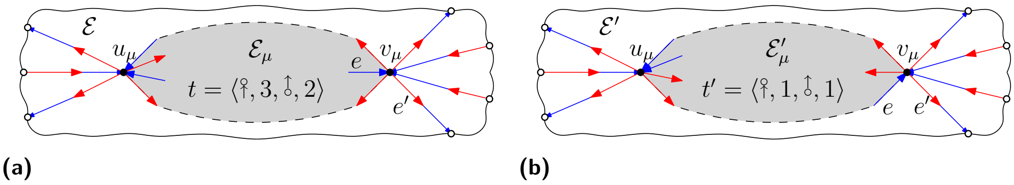
$\mathcal{E}$
u_μ
$\mathcal{E}_\mu$
$t = \langle \overset{\circ}{\uparrow}, 3, \underset{\circ}{\uparrow}, 2 \rangle$
e
v_μ
e'
(a)
$\mathcal{E}'$
u_μ
$\mathcal{E}'_\mu$
$t' = \langle \overset{\circ}{\uparrow}, 1, \underset{\circ}{\uparrow}, 1 \rangle$
v_μ
e
e'
(b)

Figure 4 Illustration for the proof of Lemma 10. The parity of t and t' is the same at u_μ and different at v_μ; in particular, even if a new alternation is introduced between the pair (e, e') at v_μ, the different parity guarantees that the modality at v_μ does not increase from $\mathcal{E}$ to $\mathcal{E}'$.

Sketch. We show how to construct a drawing Γ'_G of G satisfying m in which the embedding of pert(μ) is $\mathcal{E}'_\mu$; see Fig. 4. Let Γ_G be a drawing of G whose embedding is $\mathcal{E}$. Remove from Γ_G the drawing of all the vertices of pert(μ) different from u_μ and v_μ and the drawing of all the edges of pert(μ). Denote by f the face of the resulting embedded graph G^- that used to contain the removed vertices and edges. We obtain Γ'_G by inserting a drawing of pert(μ) whose embedding is $\mathcal{E}'_\mu$ in the interior of f so that vertices u_μ and v_μ are identified with their copies in G^-. We can prove that the embedding $\mathcal{E}'$ of Γ'_G satisfies m by exploiting the interplay between the parity and the number of alternations at u_μ (at v_μ) in t' and t when $t' \preceq t$. ◀

Let $\mathcal{T}$ be the SPQR-tree $\mathcal{T}$ of G rooted at a reference edge e. In the remainder of the section, we show how to compute the gist $S(\mu)$ of the set of good tuples for μ, for each non-root node μ of $\mathcal{T}$. In the subsequent procedures to compute $S(\mu)$ for S-, P-, and R-nodes, we are not going to explicitly avoid set $S(\mu)$ to contain dominated tuples. In fact, this can always be done at the cost of an additive $O(k^2)$ factor in the running time, by maintaining an hash table that stores the tuples that have been constructed (possibly multiple times) by the procedures and by computing the gist of the constructed set as a final step.

▶ **Property 1.** *For each node $\mu \in \mathcal{T}$, it holds that $|S(\mu)| \in O(k)$.*

Proof. By the definition of gist, any embedding pair (σ, a) has at most two tuples $t', t'' \in S(\mu)$ such that (σ, a) is the embedding pair of t' and t'' at u_μ; also, the embedding pairs (σ', a') of t' and (σ'', a'') of t'' at v_μ are such that $\sigma' \neq \sigma''$. Since there exist at most $2k$ realizable embedding pairs (σ, a) at u_μ (as $\sigma \in \{\overset{\circ}{\uparrow}, \underset{\circ}{\uparrow}\}$, $a \in \{0, 1, \dots, k\}$, and the existence of tuple whose embedding pair at u_μ is $(\sigma, 0)$ implies that all tuples have such an embedding pair at u_μ), we have $|S(\mu)| \leq 4k$. ◀

If μ is a leaf Q-node of $\mathcal{T}$, then $S(\mu) = \{\langle \sigma(u_\mu v_\mu, u_\mu), 0, \sigma(u_\mu v_\mu, v_\mu), 0 \rangle\}$. If μ is an internal node of $\mathcal{T}$, we visit $\mathcal{T}$ bottom-up and compute the set $S(\mu)$ for μ assuming to have already computed the sets $S(\mu_1), \dots, S(\mu_k)$ for the children $\mu_1, \dots, \mu_k$ of μ (where μ_i is the child of μ corresponding to the edge e_i in skel(μ)). Let ρ be the unique child of the root of $\mathcal{T}$. Once the set $S(\rho)$ has been determined, we can efficiently decide whether G admits an embedding satisfying m in which the reference edge e is incident to the outer face by means of the following lemma.

▶ **Lemma 11.** *Given $S(\rho)$, we can test whether G has an embedding that satisfies m in $O(k^2)$ time.*

5 Partial 2-trees

In the following, we describe how to compute $S(\mu)$, if μ is an S-node (Lemma 12) and a P-node (Lemma 13) in $O(f(k)|\operatorname{skel}(\mu)|)$ time, where f is a computable function.

▶ **Lemma 12.** *Set $S(\mu)$ can be constructed in $O(k^2|\operatorname{skel}(\mu)|)$ time for an S-node μ.*

Sketch. Let μ be an S-node with skeleton $\operatorname{skel}(\mu) = (e_1, e_2, \dots, e_h)$. We define τ_j as the S-node obtained by the series composition of $\mu_1, \mu_2, \dots, \mu_j$, with $j \leq h$. Initially we set $S(\tau_1) = S(e_1)$. Then, we construct $S(\tau_j)$, for $j = 2, \dots, h$, by verifying the compatibility of the embedding pairs of the good tuples of the virtual edges of $\operatorname{skel}(\tau_j)$ at the internal vertices of $\operatorname{skel}(\tau_j)$. As $S(\tau_j) = S(\tau_{j-1}) \cup e_j$, we can compute $S(\tau_j)$ by considering all the tuples obtained by combining every tuple $t' \in S(\tau_{j-1})$ with every tuple $t'' \in S(e_j)$. Since both these sets contain $O(k)$ tuples, by Property 1, and since the tuple resulting from the combination of t' and t'' can be determined in $O(1)$ time, we have that $S(\tau_j)$ can be computed in $O(k^2)$ time. Therefore, the overall running time for computing $S(\mu) = S(\tau_h)$ is $O(k^2|\operatorname{skel}(\mu)|)$. ◀

▶ **Lemma 13.** *Set $S(\mu)$ can be constructed in $O((2k+4)!k^3 + |\operatorname{skel}(\mu)|)$ time for a P-node μ.*

Sketch. Let μ be a P-node with poles u_μ and v_μ, whose skeleton $\operatorname{skel}(\mu)$ consists of h parallel virtual edges $e_1, e_2, \dots, e_h$. It can be shown that the computation of $S(\mu)$ reduces in $O(|\operatorname{skel}(\mu)|)$ time to the computation of $S(\tau)$, where τ is a P-node whose skeleton consists of at most $2k$ virtual edges of $\operatorname{skel}(\mu)$ that contribute with at least one alternating pair of edges at u_μ or v_μ, plus up to 4 virtual edges of $\operatorname{skel}(\mu)$ that contribute with no alternating pair at u_μ or at v_μ. For any permutation π of the virtual edges of $\operatorname{pert}(\tau)$, let τ_i^π be the P-node obtained by restricting τ to the first i virtual edges in π. We fix the embedding of $\operatorname{skel}(\tau_i^\pi)$ in such a way that the virtual edges of $\operatorname{skel}(\tau_i^\pi)$ are ordered according to π. Then, in a fashion similar to the S-node case, we can compute $S(\tau_i^\pi)$ for the given embedding of $\operatorname{skel}(\tau_i^\pi)$ by combining $S(\tau_{i-1}^\pi)$ and $S(e_i)$ in $O(k^2)$ time (recall that both these sets have size $O(k)$, by Property 1). Clearly, for any fixed π, we can compute $S(\tau_\pi^h)$ in $O(k^3)$ time. Thus, by performing the above computation for all the $(2k+4)!$ possible permutations for the virtual edges of $\operatorname{pert}(\tau)$, we can construct $S(\tau)$ in $O((2k+4)!k^3 + |\operatorname{skel}(\mu)|)$ time. ◀

Altogether, Lemmas 12 and 13 yield the following main result.

▶ **Lemma 14.** *k-MaxModality can be solved in $O((2k+4)!k^3n)$ for series-parallel digraphs.*

Observation 1, Lemma 14, and Theorem 9 immediately imply the following.

▶ **Corollary 15.** *k-Modality can be solved in $O(((2k+4)!k^3\log k)n)$ for directed partial 2-trees.*

Due to the special algorithmic framework we are employing, we can however turn the multiplicative $O(\log k)$ factor in the running time into an additive $O(k)$ factor by modifying Theorem 9 as follows. When considering a cut-vertex v, we will execute "only once" the function TestBiconnected by rooting the SPQ-tree at a Q-node η corresponding to an edge incident to v. This will allow us to compute the minimum modality for cut-vertex v in an embedding that satisfies m at every vertex, by simply scanning the set $S(\eta)$, which takes $O(k)$ time by Property 1, rather than by exploiting a logarithmic number of calls to TestBiconnected.

▶ **Theorem 16.** *k-Modality can be solved in $O((2k+4)!k^3n)$ for directed partial 2-trees.*

6 A Linear-time Algorithm for 4-MaxModality when $\Delta \leq 6$

In this section, we show that in the special case when $k = 4$ and G has maximum degree $\Delta \leq 6$, it is possible to compute the set $S(\mu)$ when μ is an R-node in linear time in the size of skel(μ).

Our strategy to compute $S(\mu)$ is as follows. We select a single tuple from the admissible set of each virtual edge incident to u_μ and v_μ, in every possible way. Each selection determines a "*candidate tuple*" t for $S(\mu)$. First, we check if t is admissible at both u and v. Second, we restrict the tuples of the edges incident to the poles to only the tuples that form t and check if there is a way of satisfying m at the (inner) vertices of skel(μ). If both the poles and the inner vertices are satisfiable, then we add t to $S(\mu)$. Since the degrees of the poles are bounded, there is at most a constant number of candidate tuples which must be checked. The complexity lies in this check.

We now formally describe how to compute $S(\mu)$. First, for each virtual edge e_i of skel(μ) incident to the poles of μ, we select a tuple t_i from $S(\mu_i)$. Let $T_u = [t_{u,1}, t_{u,2}, \dots, t_{u,\ell}]$ and $T_v = [t_{v,1}, t_{v,2}, \dots, t_{v,h}]$ be the list of tuples selected for the virtual edges incident to u_μ and to v_μ, respectively. Each pair of lists T_u and T_v yields a *candidate tuple* $t = \langle \sigma_1, a, \sigma_2, b \rangle$ for μ. However, the tuples selected to construct T_u and T_v allow for an admissible embedding of pert(μ) realizing tuple t *if and only if*: **(Condition 1)** tuple t satisfies m at u_μ and at v_μ, and **(Condition 2)** it is possible to select tuples for each of the remaining virtual edges of skel(μ) that satisfy m at every internal vertex of skel(μ). Let $\mathcal{P}(\mu)$ be the set of candidate tuples for μ constructed as described above. We can easily filter out the candidate tuples that do not satisfy Condition 1. In the remainder of the section, for each pair of lists T_u and T_v yielding a tuple $t \in \mathcal{P}(\mu)$, we will show how to test Condition 2 for μ in linear time. This and the fact that $|\mathcal{P}(\mu)| \in O(1)$ imply the following.

▶ **Lemma 17.** *Set $S(\mu)$ can be constructed in $O(|\operatorname{skel}(\mu)|)$ time for an R-node μ, if $\Delta \leq 6$.*

Altogether, Lemmas 12, 13 and 17 yield the following main result.

▶ **Lemma 18.** 4-MaxModality *can be solved in linear time for biconnected digraphs with $\Delta \leq 6$.*

Observation 1, Lemma 18, and Theorem 9 immediately imply the following.

▶ **Theorem 19.** 4-Modality *can be solved linear time for digraphs with $\Delta \leq 6$.*

To prove Lemma 17, we show how to solve the following auxiliary problem for special instances.

Problem: 4-MaxSkelModality

Input: A triple $\langle G = (V, E), \mathcal{S} = \{S(e_1), \dots, S(e_{|E|})\}, m \rangle$ where G is an embedded directed graph, each $S(e_i)$ is a set containing embedding tuples for the virtual edge $e_i \in E$, and $m : V \to \mathbb{E}_4^+$ is the *maximum-modality function.*
Question: Can we select a tuple from each set $S(e_i)$ in such a way that the modality at each vertex $v \in V$ is at most $m(v)$?

For each pair of lists T_u and T_v yielding a candidate tuple in $\mathcal{P}(\mu)$, we will construct an instance $I_\mu(T_u, T_v) = (G, \mathcal{S}, m)$ of 4-MaxSkelModality as follows. First, we set $G = \operatorname{skel}(\mu)$ and we fix the embedding of G to be equal to the unique regular embedding of skel(μ). Second, for each virtual edge $e_{u,i}$ incident to u_μ, with $i = 1, \dots, \ell$, we set $S(e_{u,i}) = \{t_{u,i}\}$; for each virtual edge $e_{v,j}$ incident to v_μ, with $j = 1, \dots, h$, we set $S(e_{v,j}) = \{t_{v,j}\}$; and, for each of the remaining virtual edges e_d of skel(μ), we set $S(e_d) = S(\mu_d)$. Finally, the maximum-modality function of I_μ coincides with m.

Clearly, $I_\mu(T_u, T_v)$ is a positive instance of 4-MaxSkelModality if and only if, given the constrains imposed by the tuples in T_u and in T_v, there exists a selection of tuples for the edges of G not incident to u_μ or v_μ that satisfies m at all the internal vertices of G, i.e., Condition 2 holds.

Let v be a vertex of G and let e be an edge in $E(v)$, we denote by $A_v(e)$ the maximum number of alternations at v over all the tuples in $S(e)$.

▶ **Definition 20** (Good instances). *An instance of* 4-MaxSkelModality *is* good *if, for any vertex v in G, it holds $\sum_{e \in E(v)}(A_v(e)+1) \leq 6$.*

Note that, for each edge e in *ske*(μ) incident to a vertex v, pert(e) contributes at least $A_v(e) + 1$ edges to $d_{\text{pert}(e)}(v)$. Thus, we have $\sum_{e \in E(v)}(A_v(e)+1) \leq \sum_{e \in E(v)} d_{\text{pert}(e)}(v) \leq 6$. Therefore, instance $I_\mu(T_u, T_v)$ is good. Although 4-MaxSkelModality turns out to be NP-complete in general [21, Section 9], we are now going to show the following main positive result.

▶ **Theorem 21.** 4-MaxSkelModality *is linear-time solvable for good instances.*

The outline of the linear-time algorithm to decide whether a good instance $I = \langle G = (V, E), \mathcal{S} = \{S(e_1), \dots, S(e_{|E|})\}, m\rangle$ of 4-MaxSkelModality is a positive instance is as follows.

- We process I by means of a set of *reduction rules* applied locally at the vertices of G and their incident edges. Each of these rules, if applicable, either detects that the instance I is a negative instance or transforms it into an equivalent smaller instance $I' = \langle G', \mathcal{S}', m'\rangle$. Each rule can be applied when specific conditions are satisfied at the considered vertex. A rule may additionally set a vertex as *marked*. Any marked vertex v has the main property that **any selection** of tuples from the admissible sets of the edges incident to v satisfies m' at v.
- Let I^* be the instance of 4-MaxSkelModality obtained when no reduction rule may be further applied. We prove that instance I^* has a special structure that allows us to reduce the problem of testing whether I^* is a positive instance of 4-MaxSkelModality to that of verifying the NAE-satisfiability of a constrained instance of NAESAT, in fact, of Planar NAESAT. Since Planar NAESAT is in P [25], this immediately implies that 4-MaxSkelModality is also in P. However, in [21, Section 7], by strengthening a result of Porschen et al. [26], we are able to show that the constructed instances of NAESAT are always satisfiable and that a satisfying NAE-truth assignment can be computed in linear time.

In [21, Section 8], we provide three reduction rules that turn a good instance I into an equivalent smaller good instance I'. Let $I^* = \langle G^*, \mathcal{S}^*, m^*\rangle$ be the good instance, equivalent to I, produced by applying a maximal sequence of reduction rules to I. We say that I^* is *irreducible.* The following lemma will prove useful.

▶ **Lemma 22.** *For each unmarked vertex $v \in V(G^*)$, it holds that: (i) v has degree 3, (ii) $m^*(v) = 4$, and (iii) there exist tuples $t_1, t_2 \in S^*(e)$ such that the embedding pair of t_1 and of t_2 at v are (⫯, 1) and (⫰, 1), respectively, for each edge e incident to v.*

Our next and final tool is the following, quite surprising, result.

▶ **Lemma 23.** *Any irreducible good instance I^* is a positive instance.*

Theorem 21 immediately follows from Lemma 23. We conclude the section by providing a sketch of the proof of Lemma 23. A detailed proof can be found in [21, Section 8].

Outline of the proof of Lemma 23. If a vertex is marked then any combination of tuples will satisfy m^* at it. So the proof is mainly concerned with unmarked vertices. By Lemma 22, edges where both endpoints are unmarked have one of the following tuple sets: $S_A = \{\langle \overset{\circ}{\uparrow}, 1, \overset{\circ}{\uparrow}, 1\rangle, \langle \underset{\circ}{\uparrow}, 1, \underset{\circ}{\uparrow}, 1\rangle\}$, $S_B = \{\langle \underset{\circ}{\uparrow}, 1, \overset{\circ}{\uparrow}, 1\rangle, \langle \overset{\circ}{\uparrow}, 1, \underset{\circ}{\uparrow}, 1\rangle\}$, or $S_A \cup S_B$. In the last case we arbitrarily remove either S_A or S_B. Taking advantage of the structure of irreducible instances, the problem of solving I^* is reduced in linear time to the one of testing the NAE-satisfiability of a CNF-formula ϕ in which every variable occurs in at most two clauses. Each edge incident to an unmarked vertex has the two possible embedding pairs $(\overset{\circ}{\uparrow}, 1)$ or $(\underset{\circ}{\uparrow}, 1)$ at the vertex. We create a variable for each incidence between an edge and an unmarked vertex. For each edge with two unmarked endpoints, we introduce an *edge clause* to ensure that the embedding pairs for each endpoint are selected in a consistent way. Consider an unmarked vertex v and assume, for simplicity of description, that its three incident edges have the same orientation at v. A selection of embedding pairs for the edges incident to v will not satisfy $m^*(v)$ if and only if all such pairs coincide. Therefore, we can introduce a *vertex clause* to model such constraint as a NAESAT clause that is the disjunction of the three boolean variables for the endpoints of the edges incident to v. The NAE-formula ϕ has the property that each variable occurs in at most two clauses. Moreover, the variable-clause graph G_ϕ of ϕ contains no connected component that is isomorphic to a simple cycle, since vertex clauses have degree 3. In [21, Section 7], we prove that such instances are always NAE-satisfiable and provide a linear-time algorithm to construct a NAE-truth assignment for such formulas. This proves that I^* is always a positive instance.

7 Conclusions

In this paper, we studied the complexity of the k-Modality problem, with special emphasis on $k = 4$. We provided complexity, algorithmic, and combinatorial results. Our main algorithmic contribution for $k = 4$ and $\Delta \leq 6$ leverages an elegant connection with the NAE-satisfiability of special CNF formulas, whose study allowed us to strengthen a result in [26]. Moreover, we showed notable applications of the previous results to some new interesting embedding problems for clustered networks, some of which solve open problems in this area [3, 18].

References

1 Patrizio Angelini, Giordano Da Lozzo, Marco Di Bartolomeo, Valentino Di Donato, Maurizio Patrignani, Vincenzo Roselli, and Ioannis G. Tollis. Algorithms and Bounds for L-Drawings of Directed Graphs. *Int. J. of Foundations of Computer Science*, 29(04):461–480, 2018. doi:10.1142/S0129054118410010.

2 Patrizio Angelini, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Maurizio Patrignani, and Ignaz Rutter. Beyond Level Planarity. In Yifan Hu and Martin Nöllenburg, editors, *GD '16*, volume 9801 of *LNCS*, pages 482–495. Springer, 2016. doi:10.1007/978-3-319-50106-2_37.

3 Patrizio Angelini, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Maurizio Patrignani, and Ignaz Rutter. Intersection-Link Representations of Graphs. *J. Graph Algorithms Appl.*, 21(4):731–755, 2017. doi:10.7155/jgaa.00437.

4 Patrizio Angelini, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, and Vincenzo Roselli. The importance of being proper: (In clustered-level planarity and T-level planarity). *Theor. Comput. Sci.*, 571:1–9, 2015.

5 Patrizio Angelini, Peter Eades, Seok-Hee Hong, Karsten Klein, Stephen G. Kobourov, Giuseppe Liotta, Alfredo Navarra, and Alessandra Tappini. Turning Cliques into Paths to Achieve Planarity. In Therese C. Biedl and Andreas Kerren, editors, *GD 2018*, volume 11282 of *LNCS*, pages 67–74. Springer, 2018. doi:10.1007/978-3-030-04414-5_5.

6 Patrizio Angelini, Giordano Da Lozzo, Giuseppe Di Battista, Valentino Di Donato, Philipp Kindermann, Günter Rote, and Ignaz Rutter. Windrose Planarity: Embedding Graphs with Direction-Constrained Edges. *ACM Trans. Algorithms*, 14(4):54:1–54:24, September 2018. doi:10.1145/3239561.

7 Christian Bachmaier, Franz-Josef Brandenburg, and Michael Forster. Radial Level Planarity Testing and Embedding in Linear Time. *J. Graph Algorithms Appl.*, 9(1):53–97, 2005. URL: http://jgaa.info/accepted/2005/BachmaierBrandenburgForster2005.9.1.pdf, doi:10.7155/jgaa.00100.

8 Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward Drawings of Triconnected Digraphs. *Algorithmica*, 12(6):476–497, 1994. doi:10.1007/BF01188716.

9 Carla Binucci, Walter Didimo, and Francesco Giordano. Maximum upward planar subgraphs of embedded planar digraphs. *Comput. Geom.*, 41(3):230–246, 2008.

10 Carla Binucci, Walter Didimo, and Maurizio Patrignani. Upward and quasi-upward planarity testing of embedded mixed graphs. *Theor. Comput. Sci.*, 526:75–89, 2014. doi:10.1016/j.tcs.2014.01.015.

11 Kellogg S. Booth and George S. Lueker. Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms. *J. Comput. Syst. Sci.*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1.

12 Guido Brückner and Ignaz Rutter. Partial and Constrained Level Planarity. In Philip N. Klein, editor, *SODA '17*, pages 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.

13 Steven Chaplick, Markus Chimani, Sabine Cornelsen, Giordano Da Lozzo, Martin Nöllenburg, Maurizio Patrignani, Ioannis G. Tollis, and Alexander Wolff. Planar L-Drawings of Directed Graphs. In Fabrizio Frati and Kwan-Liu Ma, editors, *GD '17*, volume 10692 of *LNCS*, pages 465–478. Springer, 2017. doi:10.1007/978-3-319-73915-1_36.

14 Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, and Maurizio Patrignani. Computing NodeTrix Representations of Clustered Graphs. *J. Graph Algorithms Appl.*, 22(2):139–176, 2018.

15 Giuseppe Di Battista and Enrico Nardelli. Hierarchies and planarity theory. *IEEE Trans. Systems, Man, and Cybernetics*, 18(6):1035–1046, 1988. doi:10.1109/21.23105.

16 Giuseppe Di Battista and Roberto Tamassia. On-Line Graph Algorithms with SPQR-Trees. In Mike Paterson, editor, *ICALP '90*, volume 443 of *LNCS*, pages 598–611. Springer, 1990. doi:10.1007/BFb0032061.

17 Ashim Garg and Roberto Tamassia. On the Computational Complexity of Upward and Rectilinear Planarity Testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.

18 Emilio Di Giacomo, Giuseppe Liotta, Maurizio Patrignani, and Alessandra Tappini. NodeTrix Planarity Testing with Small Clusters. In Fabrizio Frati and Kwan-Liu Ma, editors, *GD '17*, volume 10692 of *LNCS*, pages 479–491. Springer, 2017. doi:10.1007/978-3-319-73915-1_37.

19 Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. NodeTrix: a Hybrid Visualization of Social Networks. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1302–1309, 2007. doi:10.1109/TVCG.2007.70582.

20 John E. Hopcroft and Robert Endre Tarjan. Efficient Planarity Testing. *J. ACM*, 21(4):549–568, 1974. doi:10.1145/321850.321852.

21 Juan José Besa, Giordano Da Lozzo, and Michael T. Goodrich. Computing k-Modal Embeddings of Planar Digraphs. *CoRR*, abs/1907.01630, 2019. arXiv:1907.01630.

22 Michael Jünger, Sebastian Leipert, and Petra Mutzel. Level Planarity Testing in Linear Time. In Sue Whitesides, editor, *GD '98*, volume 1547 of *LNCS*, pages 224–237. Springer, 1998. doi:10.1007/3-540-37623-2_17.

23 Boris Klemz and Günter Rote. Ordered Level Planarity, Geodesic Planarity and Bi-Monotonicity. In Fabrizio Frati and Kwan-Liu Ma, editors, *GD '17*, volume 10692 of *LNCS*, pages 440–453. Springer, 2017. doi:10.1007/978-3-319-73915-1_34.

24 JiaWei Lu, Yuanming Zhang, Jun Xu, Gang Xiao, and Qianhui Althea Liang. Data Visualization of Web Service with Parallel Coordinates and NodeTrix. In *IEEE International Conference on Services Computing, SCC 2014, Anchorage, AK, USA, June 27 - July 2, 2014*, pages 766–773. IEEE Computer Society, 2014. doi:10.1109/SCC.2014.104.

25 B. M. E. Moret. Planar NAE3SAT is in P. *SIGACT News*, 19(2):51–54, June 1988. doi:10.1145/49097.49099.

26 Stefan Porschen, Bert Randerath, and Ewald Speckenmeyer. Linear Time Algorithms for Some Not-All-Equal Satisfiability Problems. In Enrico Giunchiglia and Armando Tacchella, editors, *SAT '03*, volume 2919 of *LNCS*, pages 172–187. Springer, 2003. doi:10.1007/978-3-540-24605-3_14.

27 Xinsong Yang, Lei Shi, Madelaine Daianu, Hanghang Tong, Qingsong Liu, and Paul M. Thompson. Blockwise Human Brain Network Visual Comparison Using NodeTrix Representation. *IEEE Trans. Vis. Comput. Graph.*, 23(1):181–190, 2017. doi:10.1109/TVCG.2016.2598472.