

Structural Parameters for Steiner Orientation

Tesshu Hanaka   

Kyushu University, Fukuoka, Japan

Michael Lampis  

Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Nikolaos Melissinos  

Computer Science Institute, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Edouard Nemery  

Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Hiroataka Ono  

Nagoya University, Japan

Manolis Vasilakis  

Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Abstract

We consider the STEINER ORIENTATION problem, where we are given as input a mixed graph $G = (V, E, A)$ and a set of k demand pairs (s_i, t_i) , $i \in [k]$. The goal is to orient the undirected edges of G in a way that the resulting directed graph has a directed path from s_i to t_i for all $i \in [k]$. We adopt the point of view of structural parameterized complexity and investigate the complexity of STEINER ORIENTATION for standard measures, such as treewidth. Our results indicate that STEINER ORIENTATION is a surprisingly hard problem from this point of view. In particular, our main contributions are the following:

1. We show that STEINER ORIENTATION is NP-complete on instances where the underlying graph has feedback vertex number 2, treewidth 2, pathwidth 3, and vertex integrity 6.
2. We present an XP algorithm parameterized by vertex cover number vc of complexity $n^{\mathcal{O}(vc^2)}$. Furthermore, we show that this running time is essentially optimal by proving that a running time of $n^{o(vc^2)}$ would refute the ETH.
3. We consider parameterizations by the number of undirected or directed edges ($|E|$ or $|A|$) and we observe that the trivial $2^{|E|}n^{\mathcal{O}(1)}$ -time algorithm for the former parameter is optimal under the SETH. Complementing this, we show that the problem admits a $2^{\mathcal{O}(|A|)}n^{\mathcal{O}(1)}$ -time algorithm.

In addition to the above, we consider the complexity of STEINER ORIENTATION parameterized by $tw + k$ (FPT), distance to clique (FPT), and $vc + k$ (FPT with a polynomial kernel).

2012 ACM Subject Classification Theory of computation $\rightarrow$ Parameterized complexity and exact algorithms

Keywords and phrases ETH, Steiner Orientation, Treewidth

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.38

Related Version Full Version: <https://arxiv.org/abs/2507.21445>

Funding Tesshu Hanaka: Partially supported by JSPS KAKENHI Grant Numbers JP21K17707, JP22H00513, JP23H04388, JP25K03077, and JST, CRONOS, Japan Grant Number JPMJCS24K2.

Michael Lampis: Partially supported by ANR project ANR-21-CE48-0022 (S-EX-AP-PE-AL).

Nikolaos Melissinos: Partially supported by Charles University projects UNCE 24/SCI/008 and PRIMUS 24/SCI/012, and by the project 25-17221S of GAČR.

Hiroataka Ono: Partially supported by JSPS KAKENHI Grant Numbers JP20H05967, JP22H00513, JP24K02898, JP25K03077, and JST, CRONOS, Japan Grant Number JPMJCS24K2.

Manolis Vasilakis: Partially supported by ANR project ANR-21-CE48-0022 (S-EX-AP-PE-AL).



© Tesshu Hanaka, Michael Lampis, Nikolaos Melissinos, Edouard Nemery, Hiroataka Ono, and Manolis Vasilakis;

licensed under Creative Commons License CC-BY 4.0

36th International Symposium on Algorithms and Computation (ISAAC 2025).

Editors: Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai; Article No. 38; pp. 38:1–38:14



Leibniz International Proceedings in Informatics

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

1 Introduction

The STEINER ORIENTATION problem is an NP-hard graph optimization problem, which involves assigning directions to the undirected edges of a mixed graph – a graph containing both directed and undirected edges – to satisfy specific connectivity requirements, given by pairs of terminals. The objective is to orient the undirected edges in such a way that there exists a directed path between each given terminal pair. Formally, the problem is defined as follows.

STEINER ORIENTATION

Instance: A mixed graph $G = (V, E, A)$ where E and A denote the set of edges and arcs in G respectively. Additionally, we are given a set of terminal pairs $\mathcal{T} = \{(s_i, t_i) \in V \times V : i \in [k]\}$ such that $s_i \neq t_i$, for all $i \in [k]$.

Goal: Determine whether there exists an orientation of E such that k directed paths $P_i = s_i \rightarrow \dots \rightarrow t_i$, $i \in [k]$ exist.

The STEINER ORIENTATION problem has been well-studied in bioinformatics, motivated by modeling protein–protein or protein–DNA interactions [9, 16, 18, 19]. Another motivation naturally arises from designing transportation networks. For example, in an urban transportation network with a mix of one-way streets (directed arcs) and two-way streets (undirected edges), one can consider the traffic control problem of deciding the direction of some streets to ensure routes from specific origins to destinations. This scenario is naturally modeled as the STEINER ORIENTATION problem.

Regarding its complexity, Arkin and Hassin show that STEINER ORIENTATION is NP-complete in general, but polynomially solvable if $k = 2$ [1]. From the perspective of parameterized complexity, Cygan et al. [7] proposed an $n^{\mathcal{O}(k)}$ -time algorithm for the number of terminal pairs k as a parameter, which shows that the problem belongs to the class XP. However, subsequent research has revealed the problem’s intractability for this parameterization; STEINER ORIENTATION is shown to be W[1]-hard when parameterized by k and cannot be solved in time $f(k) \cdot n^{o(k/\log k)}$ under ETH [17]. This lower bound was later improved to a $f(k) \cdot n^{o(k)}$ ETH-based lower bound even on planar graphs [4]. Surprisingly, Włodarczyk recently proved that the problem is not just W[1]-hard, but W[1]-complete [20]. This series of studies has almost completely characterized the complexity with respect to the parameter k . Despite this literature, to the best of our knowledge, the parameterized complexity of STEINER ORIENTATION is only studied for the number k of terminal pairs. The complexity with respect to graph parameters such as treewidth or vertex cover number has been almost entirely unstudied. This paper aims to provide the first systematic study in this direction.

1.1 Our Contribution

In this paper, we first show the para-NP-completeness of STEINER ORIENTATION for the parameters pathwidth, feedback vertex number, and vertex integrity.¹ Specifically, the problem is NP-complete on series-parallel graphs (i.e., graphs of treewidth at most 2) of vertex-deletion distance 2 to a forest of stars of size at most 4. By slightly modifying our reduction, we further obtain NP-hardness for grid graphs (which are planar and have

¹ In this paper, graph parameters refer to those of the underlying graph of G , that is, the graph obtained by replacing all arcs with undirected edges. We assume that the reader is familiar with standard parameters, but recall some of the relevant definitions and relations further below.

maximum degree 4) with bounded pathwidth. Notice that, in a sense, these results clearly delineate the frontier of polynomial-time versus NP-complete cases for most standard graph parameters: On the one hand, when we place no bound on the degree, STEINER ORIENTATION is already NP-hard on graphs which are 2 vertices away from an extremely restricted class. On the other hand, for bounded-degree graphs no such result can be obtained, as any graph that has bounded tree-depth and bounded degree actually has bounded size, so the hardness for bounded pathwidth graphs is essentially best possible.

The results above motivate us to consider more restrictive parameters. In this direction, we present an XP algorithm parameterized by vertex cover number, vc , that runs in time $n^{\mathcal{O}(vc^2)}$. Even though this improves the situation compared to the paraNP-completeness for treewidth, this complexity is rather disappointing, raising the natural question whether the square in the exponent is necessary. Our main contribution for this parameter is to answer this question by proving that an $n^{\mathcal{O}(vc^2)}$ time algorithm would refute the ETH. Along the way, we also show that STEINER ORIENTATION is fixed-parameter tractable when parameterized by distance to a clique, which can be seen as the dense analogue of vertex cover (a vertex cover of a graph is a clique modulator of its complement).

Moving in another direction, we consider the number of undirected or directed edges as parameters. Here, we show that STEINER ORIENTATION is fixed-parameter tractable (FPT) when parameterized by the number $|A|$ of directed edges. We observe that if $|A| = 0$, the problem is solvable in polynomial time as a consequence of Robbins' theorem [14]. Thus, our result generalizes this classical polynomial-time case. Here, it is natural to consider the number $|E|$ of undirected edges. Since we have only two directions for each undirected edge, STEINER ORIENTATION is clearly solvable in $2^{|E|}n^{\mathcal{O}(1)}$ time. Indeed, this running time is essentially tight, as a careful observation of the standard NP-hardness reduction for STEINER ORIENTATION [1] yields a $(2 - \varepsilon)^{|E|}n^{\mathcal{O}(1)}$ lower bound under the SETH.

Finally, we investigate the parameterized complexity with respect to the combined parameters $k + tw$ and $k + vc$. We show that STEINER ORIENTATION is FPT when parameterized by $k + tw$ by formulating it as an MSO_2 problem, which stands in contrast to the NP-hardness on series-parallel graphs. Furthermore, we give a polynomial kernel when parameterized by $k + vc$.

Overview of Techniques. Our NP-completeness proof for STEINER ORIENTATION on graphs of treewidth 2 is based on a direct reduction from a variant of 3-SAT, where we naturally use undirected edges to represent variables of the initial formula, with directions representing truth assignments (this is standard). The key new insight is the following: we can transmit information between copies of such edges, ensuring they must be oriented in a consistent way, by adding demand pairs which must be routed through *two* main hub vertices. By using appropriately oriented arcs to connect our gadgets to the two hubs we ensure that gadgets don't interfere with each other. In a sense, the take-home message of this construction is that the structure of G alone is not enough to measure the complexity of an instance, because the way that demand pairs interact with the graph can significantly complicate the structure of the instance.

The XP algorithm parameterized by vertex cover is based on a simple branching procedure: for each pair u, v of vertices of the vertex cover, guess if there is a directed path of length at most 2 from u to v in the solution and, if yes, which is (potentially) the vertex w in the middle of the path. This clearly leads to at most $n^{\mathcal{O}(vc^2)}$ guesses, and once we have fixed these decisions it is not hard to complete the solution because we can infer exactly which vertices of the cover have directed paths to each other.

Given the simplicity of the algorithm sketched above for vertex cover, it is somewhat surprising that this is essentially optimal (under the ETH). We establish this by giving a reduction from k -CLIQUE producing a STEINER ORIENTATION instance with vertex cover $\mathcal{O}(\sqrt{k})$. The key intuition is again that we can use a small number ($\mathcal{O}(\sqrt{k})$) of hubs, through which information will be routed. In particular, we set up a complete bipartite graph $K_{\sqrt{k}, \sqrt{k}}$ and for each edge of this graph we set up a gadget encoding a selection in the original instance: the number of edges of this graph is sufficient to encode all selections, but the vertex cover of the construction is determined by the number of vertices of the complete bipartite graph.

To obtain an algorithm for parameter $|A|$ we face significantly more obstacles than for parameter $|E|$ (which is trivially FPT). Our approach relies on exhaustive applications of some (standard) reduction rules for this problem, which eliminate all cycles and degree 1 vertices. We then focus on a restricted special case of STEINER ORIENTATION, where all undirected components are paths, all internal vertices of the paths are incident on no arcs, and the endpoints of each path are incident on at most one arc. We solve this restricted case using a reduction to 2-SAT. We then show, using arguments which essentially take into account that A is a feedback edge set of the underlying graph, that a simple branching algorithm can reduce any instance into $2^{\mathcal{O}(|A|)}$ instances of the (polynomial-time solvable) restricted case.

The FPT algorithm for distance to clique relies again on exhaustively applying reduction rules that eliminate cycles and then making some simple structural observations: there are few edges incident on the modulator vertices (so we can guess their orientation); and the edges contained in the clique must have a very simple structure (they form a matching). The FPT algorithm for $\text{tw} + k$ relies on a formulation of the problem in MSO_2 logic and Courcelle's theorem – we leave it as an interesting open problem to determine the best dependence for this parameterization. Finally, for the more restrictive parameter $\text{vc} + k$ we obtain a polynomial kernel via a maximum matching argument. In particular, we show that by calculating a maximum matching between pairs of vertices of the vertex cover and non-terminals in the independent set, we can identify a set of $\mathcal{O}(\text{vc}^2)$ vertices of the independent set which are sufficient to obtain a solution and delete the rest, giving a kernel of order $\mathcal{O}(\text{vc}^2 + k)$.

1.2 Related work

The STEINER ORIENTATION problem has been studied extensively. From a bioinformatics perspective, an optimization version of STEINER ORIENTATION is widely studied [9, 16, 18, 19]. In this optimization version, called MAXIMUM STEINER ORIENTATION, the goal is to find an orientation that maximizes the number of satisfied terminals pairs.

Unfortunately, MAXIMUM STEINER ORIENTATION is significantly more difficult than STEINER ORIENTATION. Medvedovsky et al. [16] show that the problem is inapproximable within a factor of $12/13$ even on undirected stars and binary trees. Elberfeld et al. then observe that the NP-hardness reduction for STEINER ORIENTATION in [1] implies the inapproximability within a factor of $7/8$ [10]. More recently, Hörsch [15] considered the maximization version of STEINER ORIENTATION but under the restriction that the set of demand pairs includes all possible demands and showed that this case is also APX-hard. Interestingly, for this version of the problem Hörsch supplies an XP algorithm for parameter $|A|$ and leaves it as an open question whether an FPT algorithm can be obtained. In this paper we provide an FPT algorithm for STEINER ORIENTATION parameterized by $|A|$, but this is somewhat orthogonal to Hörsch's question, as we are not dealing with the maximization version (which makes our problem easier) but we are also not assuming that all pairs of vertices have a demand (which makes our problem harder).

For the approximability, an $\mathcal{O}(\log n)$ -approximation is presented for undirected graphs [16]. This has been improved to a factor $\mathcal{O}(\log n / \log \log n)$ [11], and later to $\mathcal{O}(\log k / \log \log k)$ [7]. For bounded feedback vertex number graphs and bounded treewidth graphs, an $\mathcal{O}(\log n)$ -approximation algorithm and an $\mathcal{O}(\log^2 n)$ -approximation algorithm are presented [10].

From the viewpoint of parameterized complexity, Dorn et al. [9] and Roayaei [18] propose several FPT algorithms for parameters related to terminal pairs.

Regarding the parameterized approximation, Chitnis, Feldmann, and Suchý [4] show that there is no constant $\epsilon > 0$ such that MAXIMUM STEINER ORIENTATION on planar graphs admits an $(19/20 - \epsilon)$ -approximation² in time $f(k)n^{o(1)}$. Włodarczyk [20] shows that there is no $(\log k)^{o(1)}$ -approximation algorithm for MAXIMUM STEINER ORIENTATION that runs in FPT time with respect to k assuming $\text{FPT} \neq \text{W}[1]$.

2 Preliminaries

We use the standard notations in graph theory. Let $G = (V, E, A)$ be a mixed graph where V is a set of vertices, E is a set of undirected edges, and A is a set of directed edges. We denote by $\{u, v\} \in E$ an undirected edge between u and v and by $(u, v) \in A$ a directed edge from u to v .

A mixed graph G is a *mixed acyclic graph* if it has no cycle. Notice that in the context of mixed graphs a cycle is any sequence of vertices $v_1, v_2, \dots, v_p$ with $v_1 = v_p$ such that between any two consecutive vertices v_i, v_{i+1} the graph contains either the edge $\{v_i, v_{i+1}\}$ or the arc (v_i, v_{i+1}) . By definition, the subgraph induced by undirected edges in a mixed acyclic graph is a forest. Moreover, the graph obtained from a mixed acyclic graph G by contracting all the undirected edges is a directed acyclic graph. For a vertex subset $V' \subseteq V$, we denote by $G[V'] = (V', E(V'), A(V'))$ the subgraph induced by V' , where $E(V') \subseteq E$ and $A(V') \subseteq A$. A mixed path of a mixed graph is a path using both edges and arcs respecting the orientation of the arcs. For a mixed graph $G = (V, E, A)$, $\lambda(E)$ denotes an orientation of E and G_λ denotes the directed graph obtained by the orientation $\lambda(E)$.

Graph parameters. Throughout the paper we will use several structural graph parameters which will in general refer to the underlying graph of a given mixed graph G . Recall that the underlying graph is the graph obtained by replacing each arc by an edge with the same endpoints. The parameters we will mention are treewidth (tw), pathwidth (pw), vertex integrity (vi), feedback vertex set (fvs), vertex cover (vc), and distance to clique (dte). For the definitions of treewidth and pathwidth we refer the reader to [6]; fvs, vc, and dte denote the size of the smallest set of vertices whose removal leaves the graph a forest, an independent set, or a clique respectively; while a graph has vertex integrity at most k if there exists a set of vertices S whose removal results in a graph where all components have size at most $k - |S|$. It is known that four of these parameters form a hierarchy, in the sense that $\text{tw} \leq \text{pw} \leq \text{vi} \leq \text{vc} + 1$ for all graphs. We note that all these parameters are closed under vertex deletion and edge contraction, meaning that these operations can only decrease the parameter value of a graph.

² Note that in [4] the approximation ratio is defined as the value of the approximate solution divided by the value of the optimal solution, which results in ratios smaller than 1. In most other works, the ratio is instead defined as the optimal solution divided by the approximate solution, which gives values larger than 1.

2.1 Preprocessing

We present two basic polynomial-time reduction rules for STEINER ORIENTATION, which eliminate cycles and degree-1-vertices respectively. Applying these rules will never increase the values of our parameters, as the rules use edge contractions and vertex deletions, so in the rest of the paper we will mostly focus on instances where these rules have been exhaustively applied.

► **Proposition 1** ([19]). *Let $(G, \mathcal{T})$ be an instance of STEINER ORIENTATION. Let G' be a mixed graph obtained from G by contracting all the cycles and $\mathcal{T}'$ be the corresponding set of terminal pairs. Then $(G, \mathcal{T})$ is a yes-instance if and only if $(G', \mathcal{T}')$ is a yes-instance. Moreover, G' is a mixed acyclic graph and can be computed in polynomial time.*

► **Proposition 2.** *Let $(G, \mathcal{T})$ be an instance of STEINER ORIENTATION. If G has a vertex $v \in V$ of degree 1 in the underlying graph, then we can either correctly conclude that $(G, \mathcal{T})$ is a no-instance or construct a new instance $(G', \mathcal{T}')$ such that $G' = G[V \setminus \{v\}]$ and $(G, \mathcal{T})$ is a yes-instance if and only if $(G', \mathcal{T}')$ is a yes-instance.*

3 NP-hardness

Our main result in this section is to show that STEINER ORIENTATION remains NP-hard even on very restricted families of graphs. We focus on restrictions on the structure of the underlying undirected graph and show that even if the underlying graph is series-parallel and has vertex-deletion distance 2 to a forest of stars of size at most 4 the problem is still NP-complete. Note that all series-parallel graphs have treewidth at most 2 [2, Theorem 41] and are planar [3, Chapter 11.2], while the graph constructed by our reduction has feedback vertex number 2, pathwidth 3, and vertex integrity 6. Since STEINER ORIENTATION is polynomial-time solvable when the underlying graph of the input is a tree (as there exists a unique path between every pair of terminals), our result establishes a sharp dichotomy on the polynomial-time solvability of STEINER ORIENTATION with respect to the treewidth of the input graph. Moreover, by slightly modifying our reduction we further obtain NP-hardness for grid graphs of bounded pathwidth.

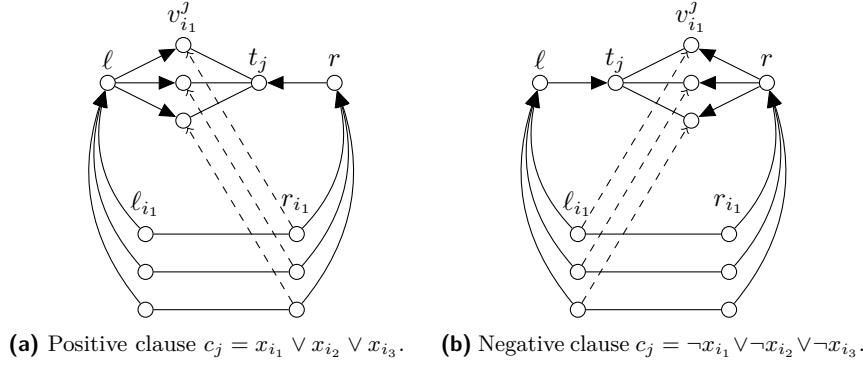
► **Theorem 3.** *STEINER ORIENTATION is NP-complete on series-parallel graphs of vertex-deletion distance 2 to a forest of stars of size at most 4.*

Proof. It is easy to see that STEINER ORIENTATION belongs to NP, thus in the rest of the proof we argue about its NP-hardness. The starting point of our reduction is the MONOTONE 3-SAT problem, which is the variation of 3-SAT where every clause contains at most 3 literals and is *monotone*, that is, it contains only unnegated or only negated variables. This variation is well-known to be NP-hard [8, 12].

Let ϕ be an instance of MONOTONE 3-SAT, where $X = \{x_1, \dots, x_n\}$ denotes its variables and $C = \{c_1, \dots, c_m\}$ its clauses. We say that a clause is *positive* if it consists of only unnegated variables, and *negative* if it consists of only negated variables.

We will construct an instance $(G, \mathcal{T})$ of STEINER ORIENTATION that is equivalent to ϕ . For each variable x_i , introduce vertices ℓ_i, r_i that are connected via an undirected edge. We further introduce vertices ℓ and r , as well as the arcs $\ell_i \rightarrow \ell$ and $r_i \rightarrow r$, for all $i \in [n]$. Now let c_j be a positive clause containing 3 variables, that is, $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$ for distinct $i_1, i_2, i_3 \in [n]$. We construct the corresponding clause gadget as depicted in Figure 1. In particular, we introduce vertices $v_{i_1}^j, v_{i_2}^j, v_{i_3}^j$, all of which have an incoming arc from ℓ , as well as a vertex t_j that is connected via undirected edges with $v_{i_1}^j, v_{i_2}^j, v_{i_3}^j$ and has an incoming arc

from r . We further introduce the terminal pairs $(\ell, t_j), (r_{i_1}, v_{i_1}^j), (r_{i_2}, v_{i_2}^j), (r_{i_3}, v_{i_3}^j)$ into $\mathcal{T}$. We treat positive clauses that contain 2 variables in an analogous way, while the construction for the negative clauses is entirely symmetric.



■ **Figure 1** The terminal pairs added during the construction of the clause gadget are denoted by dashed arcs, apart from (ℓ, t_j) and (r, t_j) respectively for the sake of clarity.

This completes the construction of the instance $(G, \mathcal{T})$ of STEINER ORIENTATION. It is easy to see that G is series-parallel, while $G - \{\ell, r\}$ is a forest whose connected components are either a K_2 , a $K_{1,2}$, or a $K_{1,3}$, thus indeed it holds that $\text{fvs}(G) = 2$, $\text{pw}(G) = 3$, and $\text{vi}(G) = 6$. It remains to argue that the instance is equivalent to ϕ .

For the forward direction, let $\alpha: X \rightarrow \{\text{true}, \text{false}\}$ be a satisfying assignment for ϕ . Consider the following orientation for the undirected edges of G . If $\alpha(x_i) = \text{true}$ then we orient the edge $\{\ell_i, r_i\}$ as $r_i \rightarrow \ell_i$, otherwise if $\alpha(x_i) = \text{false}$ then we orient it as $\ell_i \rightarrow r_i$. As for the clauses c_j that contain x_i and the edge $\{v_i^j, t_j\}$, if c_j is a positive clause (resp. negative) then if $\alpha(x_i) = \text{true}$ we orient it as $v_i^j \rightarrow t_j$ (resp. $t_j \rightarrow v_i^j$), otherwise if $\alpha(x_i) = \text{false}$ we orient it as $t_j \rightarrow v_i^j$ (resp. $v_i^j \rightarrow t_j$). It suffices to argue that this orientation satisfies all terminal pairs in $\mathcal{T}$.

Let $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$ be a positive clause involving three variables. We will argue that the described orientation satisfies all terminal pairs introduced due to the clause gadget of c_j . Since α is a satisfying assignment for ϕ , it holds that there exists $i \in \{i_1, i_2, i_3\}$ such that $\alpha(x_i) = \text{true}$. Consequently, the arc $v_i^j \rightarrow t_j$ exists in our orientation, thus the terminal pair (ℓ, t_j) is satisfied by the directed path $\ell \rightarrow v_i^j \rightarrow t_j$. As for the terminal pair $(r_{i'}, v_{i'}^j)$, where $i' \in \{i_1, i_2, i_3\}$, notice that if $\alpha(x_{i'}) = \text{true}$ then it is satisfied by the directed path $r_{i'} \rightarrow \ell_{i'} \rightarrow \ell \rightarrow v_{i'}^j$, while if $\alpha(x_{i'}) = \text{false}$ then it is satisfied by the directed path $r_{i'} \rightarrow r \rightarrow t_j \rightarrow v_{i'}^j$. The argumentation is similar for positive clauses involving two variables, and symmetric for negative clauses.

For the converse direction, assume there exists an orientation of G such that all terminal pairs in $\mathcal{T}$ are satisfied. Consider the assignment $\alpha: X \rightarrow \{\text{true}, \text{false}\}$ where $\alpha(x_i) = \text{true}$ if and only if the edge $\{\ell_i, r_i\}$ has been oriented as $r_i \rightarrow \ell_i$. We will prove by contradiction that this is a satisfying assignment for ϕ . Let $c_j = x_{i_1} \vee x_{i_2} \vee x_{i_3}$ be a positive clause involving three variables, and assume that $\alpha(x_i) = \text{false}$ for all $i \in \{i_1, i_2, i_3\}$. In that case, for the terminal pair $(r_{i'}, v_{i'}^j)$ to be satisfied, it follows that the edge $\{v_{i'}^j, t_j\}$ has been directed as $t_j \rightarrow v_{i'}^j$. Since this holds for all $i \in \{i_1, i_2, i_3\}$, it contradicts the fact that the terminal pair (ℓ, t_j) is satisfied by the orientation. The argumentation is similar for positive clauses involving two variables, and symmetric for negative clauses. ◀

► **Corollary 4.** STEINER ORIENTATION is NP-complete on grid graphs of constant pathwidth.

4 FPT algorithm by distance to clique

In this section, we give an FPT algorithm parameterized by distance to clique. More precisely, we consider the case where the underlying undirected graph of the input is at most dtc vertex deletions away from being a clique. In a sense, this parameterization is the dense analogue of the parameterization by vertex cover (because a vertex cover of a graph is a clique modulator of its complement). As usual, we will assume that the set whose removal leaves the underlying graph a clique is given to us in the input. The main result of this section is the following.

► **Theorem 5.** *STEINER ORIENTATION can be solved in time $4^{\text{dtc}} n^{\mathcal{O}(1)}$.*

We establish Theorem 5 starting via a series of reduction rules given by the lemmas below. Throughout we will assume that we are given a set S of dtc vertices such that removing S from the input graph leaves a mixed graph whose underlying graph is a clique. We will also assume that, using Proposition 1, the input is a mixed *acyclic* graph.

► **Lemma 6.** *Let G be a mixed acyclic graph and S a set of vertices such that the underlying graph of $G - S$ is a clique. Then, each vertex in S has at most one undirected edge to $V \setminus S$.*

► **Lemma 7.** *In a mixed acyclic graph G , let C be a clique in the underlying graph of G . Then there is no vertex incident to at least two undirected edges in C . In other words, the set of undirected edges in C forms a matching.*

We can now proceed to the proof of Theorem 5. The high-level idea is that by using our previous observations the number of undirected edges incident on S is at most 2dtc ; we will therefore enumerate all possible orientations of these edges and check for each orientation if it can be extended to a feasible solution. The challenge will be in establishing that this check can be performed in polynomial time.

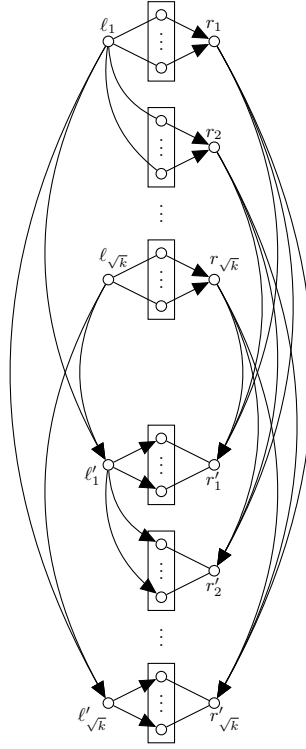
5 Vertex cover number

In this section we consider STEINER ORIENTATION parameterized by the vertex cover number vc of the input graph, arguably one of the most restrictive structural parameterizations one could consider. We first show in Theorem 8 that even in this extremely restrictive setting, the problem remains intractable and prove its $\text{W}[1]$ -hardness. As a matter of fact, our reduction implies that the problem does not admit any $n^{o(\text{vc}^2)}$ -time algorithm under the ETH, and our next result is in Theorem 10 to present such an optimal algorithm.

5.1 Hardness

► **Theorem 8.** *STEINER ORIENTATION is $\text{W}[1]$ -hard parameterized by the vertex cover number vc of the input graph, and for any computable function f it cannot be solved in time $f(\text{vc})n^{o(\text{vc}^2)}$ under the ETH.*

Proof. In k -MULTICOLORED CLIQUE we are given a graph G and a partition of $V(G)$ into k independent sets each of size n , and we are asked to determine whether G contains a k -clique. It is well-known that k -MULTICOLORED CLIQUE parameterized by k is $\text{W}[1]$ -hard and does not admit any $f(k)n^{o(k)}$ -time algorithm, where f is any computable function, unless the ETH is false [6]. Let (G, k) be an instance of k -MULTICOLORED CLIQUE, where we assume without loss of generality that $\sqrt{k} \in \mathbb{N}$ (one can do so by adding dummy independent sets connected to all the other vertices of the graph). Recall that we assume that G is given to us partitioned into k independent sets $V_1, \dots, V_k$, where $V_i = \{v_1^i, \dots, v_n^i\}$. We will construct in polynomial time an equivalent instance $(H, \mathcal{T})$ of STEINER ORIENTATION, with $\text{vc}(H) = \mathcal{O}(\sqrt{k})$.



■ **Figure 2** Part of the construction of graph H . Rectangles denote independent sets of size n .

Construction. We first introduce the vertices $\{\ell_\alpha, \ell'_\alpha, r_\alpha, r'_\alpha : \alpha \in [\sqrt{k}]\}$ and add the arcs $\ell_\alpha \rightarrow \ell'_\beta$ and $r_\alpha \rightarrow r'_\beta$ for all $\alpha, \beta \in [\sqrt{k}]$. For all $i \in [k]$, we further introduce the independent sets $X_i = \{x_1^i, \dots, x_n^i\}$ and $Y_i = \{y_1^i, \dots, y_n^i\}$. We fix a bijective function $h: [k] \rightarrow [\sqrt{k}]^2$ that maps every independent set V_i to a distinct pair $(\alpha, \beta) \in [\sqrt{k}]^2$, and for all $i \in [k]$ with $h(i) = (\alpha, \beta)$

- we add the edge $\{\ell_\alpha, x_j^i\}$ for all $x_j^i \in X_i$,
- we add the arc (x_j^i, r_β) for all $x_j^i \in X_i$,
- we add the arc (ℓ'_α, y_j^i) for all $y_j^i \in Y_i$,
- we add the edge $\{r'_\beta, y_j^i\}$ for all $y_j^i \in Y_i$.

This completes the construction of the graph H , and we refer to Figure 2 for an illustration of a part of it. As for the terminal pairs, we proceed in three steps:

1. We initially add into $\mathcal{T}$ the pairs (ℓ_α, r_β) and (ℓ'_α, r'_β) , for all $\alpha, \beta \in [\sqrt{k}]$.
2. Next, for all $i \in [k]$ and $j \in [n]$, we add the terminal pair $(x_j^i, y_{j'}^i)$ into $\mathcal{T}$, for all $j' \in [n] \setminus \{j\}$. We refer to the terminal pairs added in this step as *consistency terminal pairs*.
3. Lastly, for distinct $i_1, i_2 \in [k]$, if $\{v_{j_1}^{i_1}, v_{j_2}^{i_2}\} \notin E(G)$ we add the terminal pairs $(x_{j_1}^{i_1}, y_{j_2}^{i_2}), (x_{j_2}^{i_2}, y_{j_1}^{i_1})$ into $\mathcal{T}$, where $j_1, j_2 \in [n]$. We refer to the terminal pairs added in this step as *edge-checking terminal pairs*.

This completes the construction of the instance $(H, \mathcal{T})$.

It is easy to see that $\{\ell_\alpha, \ell'_\alpha, r_\alpha, r'_\alpha : \alpha \in [\sqrt{k}]\}$ is a vertex cover of H , thus $\text{vc}(H) = \mathcal{O}(\sqrt{k})$. To complete the proof, we show the following claim.

▷ **Claim 9.** The instance (G, k) of k -MULTICOLORED CLIQUE is a yes-instance if and only if the constructed instance $(H, \mathcal{T})$ of STEINER ORIENTATION is a yes-instance. ◀

5.2 XP algorithm

In this section, we give an XP algorithm parameterized by vc with complexity $n^{\mathcal{O}(vc^2)}$, matching our lower bound result.

► **Theorem 10.** *There is an algorithm that solves STEINER ORIENTATION in time $n^{\mathcal{O}(vc^2)}$.*

Proof. First we note that contracting all mixed cycles as in Proposition 1 does not increase the size of a minimal vertex cover, so we can suppose that our graph is mixed acyclic.

Let G be a graph of vertex cover number vc . Let $S \subseteq V$ be a vertex cover G of size vc . It can be computed in time $2^{\mathcal{O}(vc)}$ [13]. First we guess the orientation of the edges of $G[S]$. There are $2^{\mathcal{O}(vc^2)}$ possibilities. Then, for all pairs $(u, v) \in S^2$, we guess if there is, in the final solution, an oriented path $u \rightarrow w \rightarrow v$ with $w \in V \setminus S$; if we guessed there is at least one, we guess one such w , and orient its edges in consequence. There are $n^{\mathcal{O}(vc^2)}$ possibilities. As we will see in Claim 12, after this guessing phase, as S is a vertex cover, the connectivity of S in any solution respecting the guesses is already decided. This will allow us to check the existence of such a solution in polynomial time.

▷ **Claim 11.** Let $w \in V \setminus S$, $u, v \in S$ such that at the end of the guessing steps, there are two undirected edges $\{u, w\}$ and $\{v, w\}$. In any feasible solution respecting the guesses, either we orient them both from w to u, v or from u, v to w .

The above claim shows that for any $w \in V \setminus S$, after the guessing steps, either all of its undirected edges must be oriented from w to S or from S to w .

▷ **Claim 12.** Let $u, v \in V$; in any solution respecting the guesses, if there is a path from u to v , then there is one that only uses already oriented edges after the guessing steps, except possibly for the first one if $u \in V \setminus S$, and the last one if $v \in V \setminus S$.

This claim shows us that after the guessing steps, for any (s_i, t_i) pair of terminals that is not already satisfied, the only way to satisfy them while respecting the guessing steps is to orient the edges of s_i and t_i .

We now argue that there exists a demand pair (s_i, t_i) which should be treated “first”. To see this, consider an auxiliary directed graph which has a vertex for each demand pair, and an arc from (s_i, t_i) to (s_j, t_j) when $s_j = t_i$. If this auxiliary graph contains a directed cycle, this implies that the original instance must be oriented in a way that creates a directed cycle, but since we assumed that G is mixed acyclic, this is impossible. Hence, the interesting case is when the auxiliary graph is a DAG, therefore there exists a demand pair (s_i, t_i) such that for all other demands (s_j, t_j) we have, $t_j \neq s_i$.

We can apply the following method: take such a pair (s_i, t_i) (as described in the previous paragraph). If $s_i \in V \setminus S$, then orient all undirected edges between s_i and S from s_i to S . If the demand (s_i, t_i) is now satisfied, that is, there is a path from s_i to t_i using arcs and oriented edges, we remove this demand.

If (s_i, t_i) is still unsatisfied, if $t_i \in V \setminus S$, then orient all undirected edges between t_i and S from S to t_i .

We have now handled the demand (s_i, t_i) : if it is satisfied after the steps above, we remove it from the instance and continue with the remaining demands; otherwise this demand cannot be satisfied and we reject. We eat this method until there are no unsatisfied terminals.

This method is correct: indeed, from Claim 12, after the guessing steps, the satisfaction of (s_i, t_i) only depends on the orientation of the undirected edges incident on $\{s_i, t_i\} \setminus S$. Furthermore, from the same claim, if $s_i \in V \setminus S$, in any feasible solution, its undirected edges will only be used to satisfy pairs of terminals that contain s_i . But our choice of terminals

ensures that for any such pair (s_j, t_j) , $s_i \neq t_j$, so $s_i = s_j$. So in a feasible solution respecting the guessing steps, if an undirected edge of s_i is used to satisfy such a pair, it will be oriented from s_i to S , so it was safe to select this orientation. For the second step, if (s_i, t_i) is still not satisfied, then from Claim 12, the only way to satisfy it while respecting the guessing steps is, if $t_i \in V \setminus S$, to direct one undirected edge between t_i and S from S to t_i . But from Claim 11, they all have the same orientation, so it was safe to orient all of them from S to t_i . If (s_i, t_i) is still unsatisfied after these steps, then, from Claim 12, there is no way to satisfy those terminals while respecting the guessing steps.

The above method can be applied in time $n^{\mathcal{O}(1)}$, and must be applied at most n^2 times. In the end, we have an algorithm of complexity $n^{\mathcal{O}(\text{vc}^2)}$. ◀

6 Parameterized by Number of Edges or Arcs

Arguably one of the most natural ways to structurally parameterize STEINER ORIENTATION is to consider either the number of undirected edges $|E|$ or the number of arcs $|A|$ as a parameter. Our goal in this section is to investigate the complexities of these two parameterizations.

We begin with the parameterization by $|E|$, which is trivially FPT: one can simply guess the orientation of each edge and then check if this gives a valid solution. The complexity is therefore $2^{|E|}n^{\mathcal{O}(1)}$. Our contribution is to observe that this trivial algorithm is perhaps optimal, as if one could obtain an algorithm of complexity $(2 - \varepsilon)^{|E|}n^{\mathcal{O}(1)}$, then the Strong Exponential Time Hypothesis (SETH) would be false. Recall that, informally, the SETH states that there is no algorithm for SAT running faster than the 2^n brute-force algorithm enumerating all assignments. We show the lower bound by reusing a simple reduction from SAT which appeared in [1].

► **Theorem 13.** *If there exists an $\varepsilon > 0$ such that STEINER ORIENTATION can be solved in time $(2 - \varepsilon)^{|E|}n^{\mathcal{O}(1)}$, then the satisfiability of a CNF formula ϕ with n variables can be decided in time $(2 - \varepsilon)^n |\phi|^{\mathcal{O}(1)}$, hence the SETH is false.*

The main result of this section is then to establish that the parameterization by $|A|$ also gives a fixed-parameter tractable problem, and indeed that we are again able to obtain a single-exponential parameter dependence. This parameterization is significantly more challenging, so we will need to rely on structural properties obtained by any graph where $|A|$ is small after we apply some simple reduction rules. The main result we obtain is stated below.

► **Theorem 14.** STEINER ORIENTATION admits a $2^{\mathcal{O}(|A|)}n^{\mathcal{O}(1)}$ algorithm.

Our high-level strategy will be made up of two parts. First, we define a (very) restricted special case of STEINER ORIENTATION, where, notably, all undirected components are paths (and we have some additional restrictions). With those severe restrictions, this case becomes simpler. In particular, we will show that it admits a polynomial algorithm, via a reduction to 2-SAT. Second, we will show how a branching procedure can reduce any initial instance into a collection of at most $2^{6|A|} = 64^{|A|}$ instances of the restricted problem. The result will then follow by composing the two algorithms. In order to ease presentation, we start with the algorithm for the restricted case in Section 6.1 and then give the algorithm for parameter $|A|$ in Section 6.2.

6.1 A Restricted Case

We define a restricted case of STEINER ORIENTATION as follows:

► **Definition 15.** An instance $G = (V, E, A)$ of STEINER ORIENTATION is called *restricted* if we have the following: all vertices $v \in V$ of mixed degree at least 3 are not incident on any unoriented edges, only arcs.

Intuitively, a restricted instance which is also mixed acyclic is an instance where each non-trivial undirected component is a path (because the maximum degree induced by E is 2 and we have no cycles), the internal vertices of each such path have no arcs connecting them to the outside, and furthermore each endpoint of such path has only one arc. In the following, we will show that these restricted instances are polynomial time solvable.

Our strategy to handle restricted instances of STEINER ORIENTATION will be the following: first, we simplify our instance by making several observations allowing us to orient several edges deterministically. Then, we encode the obtained instance as a 2-SAT formula, by defining for each terminal in an undirected path a variable which encodes whether this vertex can reach (or be reached by) one endpoint of its path or the other in the final solution, and verifying that each terminal pair is satisfied.

► **Lemma 16.** *There is an algorithm that takes a restricted STEINER ORIENTATION instance $G = (V, E, A)$ that is mixed acyclic and decides if a solution exists in time $n^{\mathcal{O}(1)}$.*

6.2 FPT algorithm by Number of Arcs

Proof of Theorem 14. We begin by applying Proposition 1 and Proposition 2 exhaustively. This does not increase $|A|$, so we can assume that the input is mixed acyclic and has no vertices of degree at most 1. Our algorithm is now the following: take all vertices of mixed degree 3 or more, and for each possible orientation of their edges, create a new instance. We claim that we obtain at most $2^{6|A|}$ restricted instances, such that at least one of them will have a solution if and only if the original instance did.

First, we argue that if we perform this branching on all vertices of mixed degree 3 or more, we obtain an instance conforming to Definition 15. Indeed, after the orientation, all vertices of mixed degree 3 are not incident on any unoriented edges, only arcs.

In order to bound the number of instances, let us count the number of edges to orient. We will need the following lemma:

► **Claim 17.** Let $T = (V, E)$ be an undirected forest, $V_3 \subseteq V$ be the set of vertices of degree at least 3 in T and $V_1 \subseteq V$ the set of vertices of degree 1. Then, $\sum_{v \in V_3} d(v) \leq 3|V_1|$.

Armed with this claim, let us continue: take $T \subseteq G[E]$ composed of the non-trivial connected components of $G[E]$. As G is mixed acyclic, it is a forest. Furthermore, as G does not contain any leaf, each leaf of T has at least one arc. Let V_0 be the set of leaves of T with a unique arc, V_1 be the set of leaves of T with two arcs or more, V_2 the set of vertices with two edges and at least one arc, and V_3 the set of vertices with at least 3 edges. We need to orient the unoriented edges of V_1 , V_2 and V_3 , which make $|V_1| + 2|V_2| + \sum_{v \in V_3} d(v)$ edges. But, as each arc has only two endpoints, $|V_0| + 2|V_1| + |V_2| \leq 2|A|$. And, from the previous claim, $\sum_{v \in V_3} d(v) \leq 3(|V_0| + |V_1|)$. Therefore, $|V_1| + 2|V_2| + \sum_{v \in V_3} d(v) \leq 3|V_0| + 4|V_1| + 2|V_2| \leq 3(|V_0| + 2|V_1| + |V_2|) \leq 6|A|$; so we need to orient at most $6|A|$ edges, thus obtaining at most $2^{6|A|}$ restricted instances. ◀

7 Kernelization

In this section we show that any instance $(G, \mathcal{T})$ of STEINER ORIENTATION admits a polynomial kernel with respect to parameter $\text{vc} + k$, where vc is the size of a minimum vertex cover of G and k is the number of terminal pairs.

In particular we will prove that:

► **Theorem 18.** STEINER ORIENTATION admits a kernel of order $\mathcal{O}(\text{vc}^2 + k)$, where vc is the size of the minimum vertex cover of G and k is the number of terminal pairs.

The high-level idea is that in the final kernel we will keep (i) the vertices of the vertex cover (which are $\mathcal{O}(\text{vc})$) (ii) all terminals (which are $\mathcal{O}(k)$) (iii) $\mathcal{O}(\text{vc}^2)$ additional vertices from the independent set. Similarly to the XP algorithm for parameter vertex cover, we use the idea that once we have fixed all paths of length at most 2 between vertices of the vertex cover, this is sufficient to fully determine connectivity in the graph. Therefore, at most $\mathcal{O}(\text{vc}^2)$ vertices from the independent set are needed to form the solution. The challenge is then to identify a set of that size which is always sufficient and we achieve this via a reduction to MAXIMUM MATCHING.

8 FPT by Treewidth plus Number of Terminals

Our goal in this section is to show that even though STEINER ORIENTATION is NP-hard for graphs of constant treewidth and also $W[1]$ -complete parameterized by the number of terminals k , it is in fact FPT when parameterized by the two parameters together. In order to ease presentation we will avoid giving an argument based on a dynamic programming algorithm and will prefer to instead rely on Courcelle's theorem [5]. This will of course have the disadvantage that we obtain an algorithm with running time of the form $f(\text{tw}, k)n^{\mathcal{O}(1)}$, where the function f is not explicitly bounded (and could be a tower of exponentials). This is sufficient for our purpose of classifying the complexity of this case as FPT, but we leave it as an interesting open problem to determine the best parameter dependence f one can achieve. We in fact establish the following slightly stronger claim:

► **Theorem 19.** STEINER ORIENTATION is FPT parameterized by the treewidth of the augmented underlying graph, that is, the treewidth of the graph obtained by taking the underlying graph of the input and adding for each terminal pair (s_i, t_i) the edge $s_i t_i$ to the graph.

We observe that Theorem 19 implies that STEINER ORIENTATION is FPT parameterized by treewidth plus k (the number of terminals), because adding the extra edges to the underlying graph can increase its treewidth by at most k .

9 Conclusion

Even though we have characterized the complexity of STEINER ORIENTATION for all possible values of treewidth (the only polynomial case is when the input is a forest), our NP-completeness only applies for graphs of pathwidth 3: is the problem in P or NP-complete for graphs of pathwidth 2. A similar question can be asked for feedback vertex set, where we show hardness when the feedback vertex set has size at least 2: does the problem become tractable on graphs which are one vertex away from being forests? On the parameterized complexity side, even though we obtain a $2^{\mathcal{O}(|A|)}n^{\mathcal{O}(1)}$ algorithm, the base in the exponential is large (64), so improving this is a natural question.

References

- 1 Esther M. Arkin and Refael Hassin. A note on orientations of mixed graphs. *Discret. Appl. Math.*, 116(3):271–278, 2002. doi:10.1016/S0166-218X(01)00228-1.
- 2 Hans L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.

- 3 Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. SIAM, 1999.
- 4 Rajesh Chitnis, Andreas Emil Feldmann, and Ondrej Suchý. A tight lower bound for planar steiner orientation. *Algorithmica*, 81(8):3200–3216, 2019. doi:10.1007/S00453-019-00580-X.
- 5 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.
- 6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 7 Marek Cygan, Guy Kortsarz, and Zeev Nutov. Steiner forest orientation problems. *SIAM J. Discret. Math.*, 27(3):1503–1513, 2013. doi:10.1137/120883931.
- 8 Andreas Darmann and Janosch Döcker. On simplified NP-complete variants of Monotone 3-Sat. *Discret. Appl. Math.*, 292:45–58, 2021. doi:10.1016/J.DAM.2020.12.010.
- 9 Britta Dorn, Falk Hüffner, Dominikus Krüger, Rolf Niedermeier, and Johannes Uhlmann. Exploiting bounded signal flow for graph orientation based on cause-effect pairs. *Algorithms Mol. Biol.*, 6:21, 2011. doi:10.1186/1748-7188-6-21.
- 10 Michael Elberfeld, Danny Segev, Colin R. Davidson, Dana Silverbush, and Roded Sharan. Approximation algorithms for orienting mixed graphs. *Theor. Comput. Sci.*, 483:96–103, 2013. doi:10.1016/J.TCS.2012.03.044.
- 11 Iftah Gamzu and Moti Medina. Improved approximation for orienting mixed graphs. *Algorithmica*, 74(1):49–64, 2016. doi:10.1007/S00453-014-9932-2.
- 12 E. Mark Gold. Complexity of automaton identification from given data. *Inf. Control.*, 37(3):302–320, 1978. doi:10.1016/S0019-9958(78)90562-4.
- 13 David G. Harris and N. S. Narayanaswamy. A faster algorithm for vertex cover parameterized by solution size. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, March 12-14, 2024, Clermont-Ferrand, France*, volume 289 of *LIPICs*, pages 40:1–40:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.STACS.2024.40.
- 14 Rafael Hassin and Nimrod Megiddo. On orientations and shortest paths. *Linear Algebra and its Applications*, 114-115:589–602, 1989.
- 15 Florian Hörsch. Maximum reachability orientation of mixed graphs, 2025. doi:10.48550/arXiv.2506.16171.
- 16 Alexander Medvedovsky, Vineet Bafna, Uri Zwick, and Roded Sharan. An algorithm for orienting graphs based on cause-effect pairs and its applications to orienting protein networks. In *Algorithms in Bioinformatics, 8th International Workshop, WABI 2008. Proceedings*, volume 5251 of *Lecture Notes in Computer Science*, pages 222–232. Springer, 2008. doi:10.1007/978-3-540-87361-7_19.
- 17 Marcin Pilipczuk and Magnus Wahlström. Directed multicut is $W[1]$ -hard, even for four terminal pairs. *ACM Trans. Comput. Theory*, 10(3):13:1–13:18, 2018. doi:10.1145/3201775.
- 18 Mehdy Roayaei. On the parameterized complexity of the problem of inferring protein-protein interaction directions based on cause-effect pairs. *Netw. Model. Anal. Health Informatics Bioinform.*, 8(1):11, 2019. doi:10.1007/S13721-019-0194-4.
- 19 Dana Silverbush, Michael Elberfeld, and Roded Sharan. Optimally orienting physical networks. *J. Comput. Biol.*, 18(11):1437–1448, 2011. doi:10.1089/CMB.2011.0163.
- 20 Michał Włodarczyk. Parameterized inapproximability for steiner orientation by gap amplification. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, volume 168 of *LIPICs*, pages 104:1–104:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.104.