A tight lower bound for steiner orientation

Chitnis, Rajesh; Feldmann, Andreas Emil

DOI:
10.1007/978-3-319-90530-3_7

A Tight Lower Bound for Steiner Orientation

Rajesh Chitnis[1] ⋆ and Andreas Emil Feldmann[2] ⋆⋆

[1] Department of Computer Science, University of Warwick, UK.
Email: rajeshchitnis@gmail.com

[2] Charles University in Prague, Czechia. Email: feldmann.a.e@gmail.com

Abstract. In the STEINER ORIENTATION problem, the input is a mixed graph G (it has both directed and undirected edges) and a set of k terminal pairs $\mathscr{T}$. The question is whether we can orient the undirected edges in a way such that there is a directed $s \leadsto t$ path for each terminal pair $(s,t) \in \mathscr{T}$. Arkin and Hassin [DAM '02] showed that the STEINER ORIENTATION problem is NP-complete. They also gave a polynomial time algorithm for the special case when $k = 2$.
From the viewpoint of exact algorithms, Cygan, Kortsarz and Nutov [ESA '12, SIDMA '13] designed an XP algorithm running in $n^{O(k)}$ time for all $k \geq 1$. Pilipczuk and Wahlström [SODA '16] showed that the STEINER ORIENTATION problem is W[1]-hard parameterized by k. As a byproduct of their reduction, they were able to show that under the Exponential Time Hypothesis (ETH) of Impagliazzo, Paturi and Zane [JCSS '01] the STEINER ORIENTATION problem does not admit an $f(k) \cdot n^{o(k/\log k)}$ algorithm for any computable function f. That is, the $n^{O(k)}$ algorithm of Cygan et al. is almost optimal.
In this paper, we give a short and easy proof that the $n^{O(k)}$ algorithm of Cygan et al. is asymptotically optimal, even if the input graph has genus 1. Formally, we show that the STEINER ORIENTATION problem is W[1]-hard parameterized by the number k of terminal pairs, and, under ETH, cannot be solved in $f(k) \cdot n^{o(k)}$ time for any function f even if the underlying undirected graph has genus 1. We give a reduction from the GRID TILING problem which has turned out to be very useful in proving W[1]-hardness of several problems on planar graphs. As a result of our work, the main remaining open question is whether STEINER ORIENTATION admits the "square-root phenomenon" on planar graphs (graphs with genus 0): can one obtain an algorithm running in time $f(k) \cdot n^{O(\sqrt{k})}$ for PLANAR STEINER ORIENTATION, or does the lower bound of $f(k) \cdot n^{o(k)}$ also translate to planar graphs?

1 Introduction

In the STEINER ORIENTATION problem, the input is a mixed graph $G = (V,E)$ (it has both directed and undirected edges) and a set of terminal pairs $\mathscr{T} \subseteq V \times V$. The question is whether we can orient the undirected edges in a way such that there is a directed $s \leadsto t$ path for each terminal pair $(s,t) \in \mathscr{T}$.

⋆ Supported by ERC grant CoG 647557 "Small Summaries for Big Data". Part of this work was done when the author was at the Weizmann Institute of Science (and supported by Israel Science Foundation grant #897/13), and visiting Charles University in Prague, Czechia.

⋆⋆ Supported by project CE-ITI (GAČR no. P202/12/G061) of the Czech Science Foundation.

STEINER ORIENTATION
Input: A mixed graph G, and a set $\mathscr{T}$ of k terminal pairs
Question: Is there an orientation of the undirected egdes of G such that the resulting graph has an $s \rightsquigarrow t$ path for each $(s,t) \in \mathscr{T}$
Parameter: k

Hassin and Megiddo [8] showed that STEINER ORIENTATION is polynomial time solvable if the input graph G is completely undirected, i.e., has no directed edges. If the input graph G is actually mixed then Arkin and Hassin [1] showed that STEINER ORIENTATION is NP-complete. They also gave a polynomial time algorithm for the special case when $k = 2$. Cygan, Kortsarz and Nutov [7] generalized this by giving an $n^{O(k)}$ algorithm for all $k \geq 1$, i.e., STEINER ORIENTATION is in XP parameterized by k. Although the algorithm of Cygan et al. is polynomial time for fixed k, the degree of the polynomial changes as k changes. This left open the question of whether one could design an FPT algorithm for STEINER ORIENTATION parameterized by k, i.e., an algorithm which runs in time $f(k) \cdot n^{O(1)}$ for some computable function f independent of n.

Pilipczuk and Wahlström [17] answered this question negatively by showing that STEINER ORIENTATION is W[1]-hard parameterized by k. As a byproduct of their reduction, they were able to show that under the Exponential Time Hypothesis (ETH) of Impagliazzo, Paturi and Zane [9,10] the STEINER ORIENTATION problem does not admit a $f(k) \cdot n^{o(k/\log k)}$ time algorithm for any computable function f. That is, the $n^{O(k)}$ algorithm of Cygan et al. is almost asymptotically optimal. This left open the following two questions:

– Can we close the gap between the $n^{O(k)}$ algorithm and the $f(k) \cdot n^{o(k/\log k)}$ hardness for STEINER ORIENTATION on general graphs?
– Is STEINER ORIENTATION FPT on planar graphs, or can we obtain an improved runtime such as $f(k) \cdot n^{O(\sqrt{k})}$?

In this paper, we answer the first question completely and make partial progress towards the second question. Formally, we show that:

Theorem 1. *The* STEINER ORIENTATION *problem is W[1]-hard parameterized by the number k of terminal pairs, even if the underlying undirected graph of the input graph has genus* 1*. Moreover, under ETH,* STEINER ORIENTATION *(on graphs of genus* 1*) cannot be solved in $f(k) \cdot n^{o(k)}$ time for any function f.*

Note that Theorem 1 only leaves open the case of graphs with genus 0, i.e., planar graphs. The open question is whether STEINER ORIENTATION admits the "square-root phenomenon" on planar graphs, i.e., can one obtain a $f(k) \cdot n^{O(\sqrt{k})}$ time algorithm[3] for PLANAR STEINER ORIENTATION, or does the lower bound of $f(k) \cdot n^{o(k)}$ also translate to planar graphs? To the best of our knowledge, even the NP-hardness of PLANAR STEINER ORIENTATION is not known.

[3] Or even an FPT algorithm

Our reduction uses some ideas given by Pilipczuk and Wahlström [17], who obtained a lower bound of $f(k) \cdot n^{o(\sqrt{k})}$ via a rather involved reduction from MULTICOLORED CLIQUE. This was later [16] improved to $f(k) \cdot n^{o(k/\log k)}$ via the standard trick of reducing from the COLORED SUBGRAPH ISOMORPHISM problem [12] instead. To obtain our tight lower bound in Theorem 1 for genus 1 graphs, we use some of the gadgets provided by Pilipczuk and Wahlström [17], but instead give a reduction from the GRID TILING problem introduced by Marx [11]. This way we obtain a cleaner and arguably simpler proof than the one given in [17]. The GRID TILING problem is defined as follows, where we use the standard notation $[n] = \{1, 2, \ldots, n\}$.

> $k \times k$ **GRID TILING**
> *Input* : Integers k, n, and k^2 non-empty sets $S_{i,j} \subseteq [n] \times [n]$ where $i, j \in [k]$.
> *Question*: For each $1 \leq i, j \leq k$ does there exist a value $s_{i,j} \in S_{i,j}$ such that
>
> – if $s_{i,j} = (x, y)$ and $s_{i,j+1} = (x', y')$ then $x = x'$, and
> – if $s_{i,j} = (x, y)$ and $s_{i+1,j} = (x', y')$ then $y = y'$.

We denote an instance of GRID TILING by $(k, n, \{S_{i,j}\}_{1 \leq i,j \leq k})$. The GRID TILING problem has turned out to be a convenient starting point for parameterized reductions for problems on planar graphs, and has been used recently in several W[1]-hardness proofs [3,4,5,13,14,15]. Under ETH, it was shown by Chen et al. [2] that k-CLIQUE[4] does not admit an algorithm running in time $f(k) \cdot n^{o(k)}$ for any function f. There is a simple reduction (see Theorem 14.28 from [6]) from k-CLIQUE to $k \times k$ GRID TILING implying the same runtime lower bound for the latter problem.

2 The reduction

We begin with describing the reduction from an instance of $k \times k$ GRID TILING to an instance of STEINER ORIENTATION with $O(k)$ terminal pairs. We will then prove that a solution to the GRID TILING instance implies a solution to STEINER ORIENTATION in the constructed instance. To finalize the proof of Theorem 1 we then prove the reverse implication as well.

2.1 Construction

Consider an instance $I = (k, n, \{S_{i,j}\}_{1 \leq i,j \leq k})$ of GRID TILING. We now build an instance $(G, \mathscr{T})$ of STEINER ORIENTATION as follows (refer to Figure 1).

– We first fix the Origin as marked in black[5].
– The "horizontal right" direction is viewed as the positive X axis and the "vertical upward" is viewed as the positive Y axis.
– **Black Grid Edges**: For each $1 \leq i, j \leq k$ we introduce the $n \times n$ grid $G_{i,j}$ to correspond to the set $S_{i,j}$ of the GRID TILING instance. In Figure 1 we highlight the gadget $G_{i,j}$ by a dotted rectangle.

[4] The k-CLIQUE problem asks whether there is a clique of size $\geq k$.
[5] This is the unique vertex which has incoming edge from b_1^1 and an outgoing edge to g_1^1

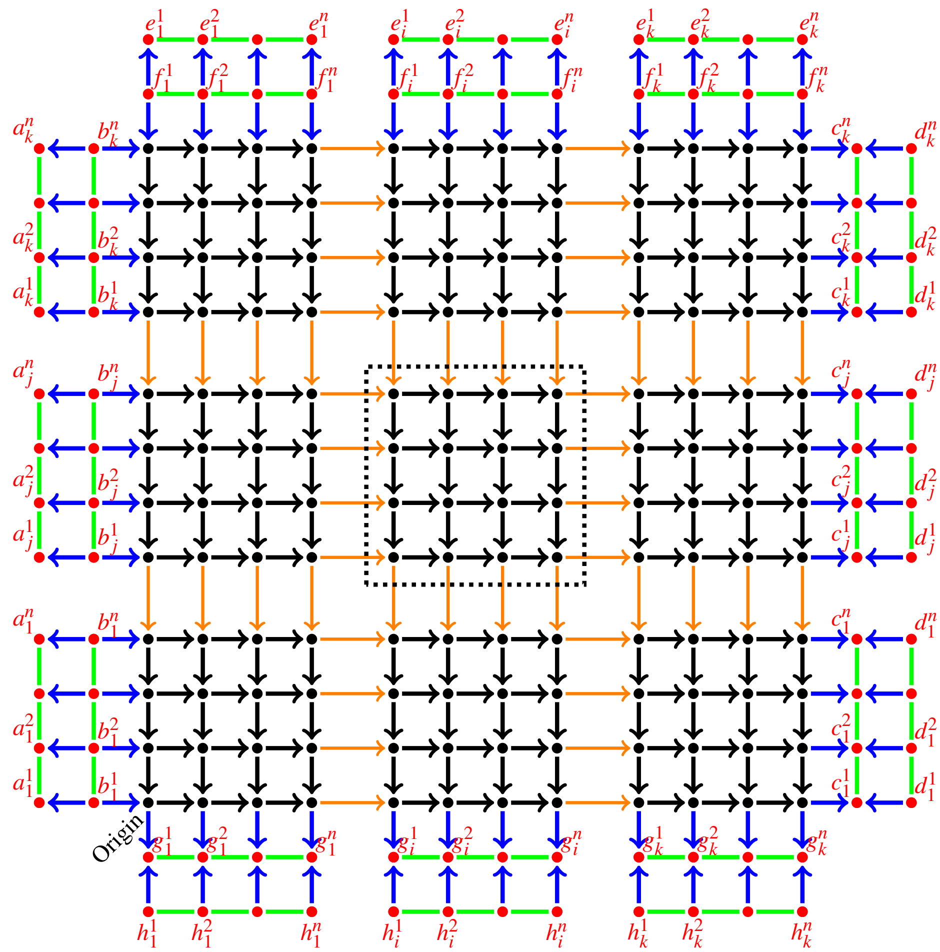

Fig. 1. The instance of STEINER ORIENTATION created from an instance of GRID TILING (before the splitting operation). At this point, the only undirected edges are the green edges. For clarity, we do not show the (directed) perfect matching (which we denote by yellow edges) given by $d_i^j \to a_i^j$ and $h_i^j \to e_i^j$ for each $i \in [k], j \in [n]$. The gadget $G_{i,j}$ is highlighted by a dotted rectangle.

- The bottom left vertex of gadget $G_{i,j}$ is denoted by $v_{i,j}^{1,1}$.
- Each row of $G_{i,j}$ is horizontal and the number of the row increases as we go vertically upwards. Similarly, each column of $G_{i,j}$ is vertical and the number of the column increases as we go horizontally rightwards. For each $1 \leq h, \ell \leq n$ the unique vertex which is the intersection of the h^{th} column and ℓ^{th} row is denoted by $v_{i,j}^{h,\ell}$.

Orient each horizontal edge of the grid $G_{i,j}$ to the right, and each vertical edge to the bottom.

– We now define four special sets of vertices for the gadget $G_{i,j}$ given by
  - $\texttt{Left}(G_{i,j}) = \{v_{i,j}^{1,\ell} \ : \ \ell \in [n]\}$
  - $\texttt{Right}(G_{i,j}) = \{v_{i,j}^{n,\ell} \ : \ \ell \in [n]\}$
  - $\texttt{Top}(G_{i,j}) = \{v_{i,j}^{\ell,n} \ : \ \ell \in [n]\}$
  - $\texttt{Bottom}(G_{i,j}) = \{v_{i,j}^{\ell,1} \ : \ \ell \in [n]\}$

– **Horizontal Orange Inter-Grid Edges**: For each $1 \leq i \leq k-1, 1 \leq j \leq k$
  - Add the directed perfect matching from vertices of $\texttt{Right}(G_{i,j})$ to $\texttt{Left}(G_{i+1,j})$ given by the set of edges $\{v_{i,j}^{n,\ell} \to v_{i+1,j}^{1,\ell} \ : \ \ell \in [n]\}$.

– **Vertical Orange Inter-Grid Edges**: For each $2 \leq j \leq k, 1 \leq i \leq k$
  - Add the directed perfect matching from vertices of $\texttt{Bottom}(G_{i,j})$ to $\texttt{Top}(G_{i,j-1})$ given by the set of edges $\{v_{i,j}^{\ell,1} \to v_{i,j-1}^{\ell,n} \ : \ \ell \in [n]\}$.

– We introduce $8k \cdot n$ red vertices given by
  - $A := \{a_i^j \mid i \in [k], j \in [n]\}$
  - $B := \{b_i^j \mid i \in [k], j \in [n]\}$
  - $C := \{c_i^j \mid i \in [k], j \in [n]\}$
  - $D := \{d_i^j \mid i \in [k], j \in [n]\}$
  - $E := \{e_i^j \mid i \in [k], j \in [n]\}$
  - $F := \{f_i^j \mid i \in [k], j \in [n]\}$
  - $G := \{g_i^j \mid i \in [k], j \in [n]\}$
  - $H := \{h_i^j \mid i \in [k], j \in [n]\}$

– **Blue Edges**:
  - For each $i \in [k], j \in [n]$
    * add the directed edge $h_i^j \to g_i^j$,
    * add the directed edge $v_{i,1}^{j,1} \to g_i^j$,
    * add the directed edge $f_i^j \to e_i^j$,
    * add the directed edge $f_i^j \to v_{i,k}^{j,n}$.
  - For each $i \in [k], j \in [n]$
    * add the directed edge $d_i^j \to c_i^j$,
    * add the directed edge $v_{k,i}^{n,j} \to c_i^j$,
    * add the directed edge $b_i^j \to a_i^j$,
    * add the directed edge $b_i^j \to v_{1,i}^{1,j}$.

– **Yellow Edges** (these are left out in Figure 1): For each $i \in [k], j \in [n]$
  - Category I: add the directed edge $d_i^j \to a_i^j$,
  - Category II: add the directed edge $h_i^j \to e_i^j$.

– **Green Edges**: For each $i \in [k]$
  - add the undirected path $a_i^1 - a_i^2 - a_i^3 - \ldots\ldots - a_i^{n-1} - a_i^n$, and denote this path[6] by A_i,
  - add the undirected path $b_i^1 - b_i^2 - b_i^3 - \ldots\ldots - b_i^{n-1} - b_i^n$, and denote this path by B_i,
  - add the undirected path $c_i^1 - c_i^2 - c_i^3 - \ldots\ldots - c_i^{n-1} - c_i^n$, and denote this path by C_i,
  - add the undirected path $d_i^1 - d_i^2 - d_i^3 - \ldots\ldots - d_i^{n-1} - d_i^n$, and denote this path by D_i,
  - add the undirected path $e_i^1 - e_i^2 - e_i^3 - \ldots\ldots - e_i^{n-1} - e_i^n$, and denote this path by E_i,
  - add the undirected path $f_i^1 - f_i^2 - f_i^3 - \ldots\ldots - f_i^{n-1} - f_i^n$, and denote this path by F_i,
  - add the undirected path $g_i^1 - g_i^2 - g_i^3 - \ldots\ldots - g_i^{n-1} - g_i^n$, and denote this path by G_i,
  - add the undirected path $h_i^1 - h_i^2 - h_i^3 - \ldots\ldots - h_i^{n-1} - h_i^n$, and denote this path by H_i.

– For each $1 \leq i, j \leq k$ and each $1 \leq x, y \leq n$ we perform the following operation on the vertex $v_{i,j}^{x,y}$:
  - If $(x,y) \in S_{i,j}$ then we keep the vertex $v_{i,j}^{x,y}$ as is.
  - Otherwise we **split** the vertex $v_{i,j}^{x,y}$ into two vertices $v_{i,j,\mathrm{LB}}^{x,y}$ and $v_{i,j,\mathrm{TR}}^{x,y}$. Note that $v_{i,j}^{x,y}$ had 4 incident edges: two incoming (one each from the left and the top) and two outgoing (one each to the right and the bottom). We change the edges as follows (see Figure 2):
    * Make the left incoming edge and bottom outgoing edge incident on $v_{i,j,\mathrm{LB}}^{s,t}$ (denoted by red color in Figure 2).
    * Make the top incoming edge and right outgoing edge incident on $v_{i,j,\mathrm{TR}}^{s,t}$ (denoted by blue color in Figure 2).
    * Add an **undirected** edge between $v_{i,j,\mathrm{LB}}^{s,t}$ and $v_{i,j,\mathrm{TR}}^{s,t}$ (denoted by the dotted edge in Figure 2).

– The set $\mathscr{T}$ of terminal pairs are given by
  - **Type I**: $(b_j^n, a_j^1), (b_j^1, a_j^n), (d_j^n, c_j^1)$ and (d_j^1, c_j^n) for each $j \in [k]$
  - **Type II**: $(f_j^n, e_j^1), (f_j^1, e_j^n), (h_j^n, g_j^1)$ and (h_j^1, g_j^n) for each $j \in [k]$
  - **Type III**: (d_j^n, a_j^1) and (d_j^1, a_j^n) for each $j \in [k]$
  - **Type IV**: (h_j^n, e_j^1) and (h_j^1, e_j^n) for each $j \in [k]$
  - **Type V**: (b_j^1, c_j^n) and (b_j^n, c_j^1) for each $j \in [k]$
  - **Type VI**: (f_j^1, g_j^n) and (f_j^n, g_j^1) for each $j \in [k]$

  Note that the total number of terminal pairs is $16k$.

Remark 1. Note that the graph G constructed above can be drawn on the surface of a torus without any two edges crossing: removing the yellow edges, the graph is clearly planar (as depicted in Figure 1) and can be drawn on a square polygon. Identifying the right edge R and left edge L of the square such that the lower left corner equals the

[6] Sometimes we also abuse notation slightly and use A_i to denote this set of vertices

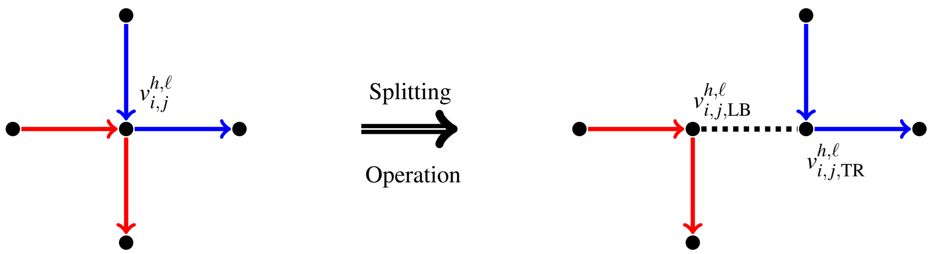

Fig. 2. The splitting operation for vertex $v^{h,\ell}_{i,j}$ when $(h,\ell) \notin S_{i,j}$. The idea behind this splitting is that no matter which way we orient the undirected dotted edge we cannot go **both** from left to right and from top to bottom. However, if we just want to go from left to right (top to bottom) then it is possible by orienting the dotted edge to the right (left), respectively.

lower right corner, and also the square's top T and bottom B edges such that the upper left corner equals the lower left corner, gives an orientable surface of genus 1 (i.e. a torus). The horizontal yellow edges of Category I can connect through $L = R$, and the vertical yellow edges of Category II can connect through $T = B$, without any edges crossing.

Remark 2. For simplicity, we add two "dummy" indices 1 and n which do not belong to any of the sets in the GRID TILING instance. Hence no vertices on the boundary of the grids $G_{i,j}$ (for any $1 \leq i, j \leq k$) are split.

Before proving the correctness of the reduction, we first introduce some notation concerning orientations of the green edges (i.e. potential solutions) of the instance.

Definition 1. *For any $i \in [n]$, a path on n vertices $a_1 - a_2 - \ldots\ldots - a_n$ is said to be oriented towards (away from) i if every edge $a_{j-1} - a_j$ is oriented towards (away from) a_j for every $j \leq i$ and every edge $a_j - a_{j+1}$ is oriented towards (away from) a_j for every $j \geq i$, respectively.*

2.2 GRID TILING has a solution $\Rightarrow$ STEINER ORIENTATION has a solution

Suppose that the instance $I = (k, n, \{S_{i,j}\}_{1 \leq i,j \leq k})$ of GRID TILING has a solution, i.e., for each $1 \leq i, j \leq k$ there exists an element $s_{i,j} \in S_{i,j}$ such that

- if $s_{i,j} = (x_{i,j}, y_{i,j})$ and $s_{i,j+1} = (x_{i,j+1}, y_{i,j+1})$ then $x_{i,j} = x_{i,j+1}$,
- if $s_{i,j} = (x_{i,j}, y_{i,j})$ and $s_{i+1,j} = (x_{i+1,j}, y_{i+1,j})$ then $y_{i,j} = y_{i+1,j}$.

That is, there exist elements $\alpha_1, \alpha_2, \ldots, \alpha_k$ and $\beta_1, \beta_2, \ldots, \beta_k$ such that for each $1 \leq i, j \leq k$ we have $(\alpha_i, \beta_j) = s_{i,j} \in S_{i,j}$. We now show that the instance $(G, \mathscr{T})$ of STEINER ORIENTATION has a solution as well. Orient the undirected green edges as follows (note that α_i and β_i are elements from $[n]$ and therefore represent row and column indices in the gadget $G_{i,j}$). For each $i \in [k]$

– orient A_i and C_i away from β_i,
– orient B_i and D_i towards β_i,
– orient E_i and G_i away from α_i,
– orient F_i and H_i towards α_i.

It is easy to see that the above orientations ensure that all terminal pairs of **Types I-IV** are satisfied. We now show that terminal pairs of **Type V** and **Type VI** are also satisfied. First we need some definitions:

Definition 2. ***(horizontal canonical paths)*** *Fix $j \in [k]$. For $1 \le \ell \le n$, we denote by Q_j^ℓ the unique (horizontal) directed $b_j^\ell \to c_j^\ell$ path whose second vertex is $v_{1,j}^{1,\ell}$ and second-last vertex is $v_{k,j}^{n,\ell}$. This path starts with the blue edge $(b_j^\ell, v_{1,j}^{1,\ell})$ and ends with the blue edge $(v_{k,j}^{n,\ell}, c_j^\ell)$. The intermediate edges are obtained by selecting the paths of black edges given by the ℓ^{th} rows of each gadget $G_{i,j}$ for $i \in [k]$, and connecting these small paths by horizontal orange edges.*

However, we need to address what to do when we encounter a split vertex on this path. Consider the vertex $v_{i,j}^{r,\ell}$ for some $i \in [k]$ and $r \in [n]$. If $v_{i,j}^{r,\ell}$ is not split, then we don't have to do anything. Otherwise, if $v_{i,j}^{r,\ell}$ is split then we add the edge $v_{i,j,LB}^{r,\ell} \to v_{i,j,TR}^{r,\ell}$ to Q_j^ℓ.

Note that the orientation of G which orients all dotted edges rightwards, i.e., $LB \to TR$, contains each of the horizontal canonical paths defined above.

Definition 3. ***(vertical canonical paths)*** *Fix $i \in [k]$. For $1 \le \ell \le n$, we denote by P_i^ℓ the unique (vertical) $f_i^\ell \to g_i^\ell$ path whose second vertex is $v_{i,k}^{\ell,n}$ and second-last vertex is $v_{i,1}^{\ell,1}$. This path starts with the blue edge $(f_i^\ell, v_{i,k}^{\ell,n})$ and ends with the blue edge $(v_{i,1}^{\ell,1}, g_i^\ell)$. The intermediate edges are obtained by selecting the paths of black edges given by the ℓ^{th} columns of each gadget $G_{i,j}$ for $j \in [k]$, and connecting these small paths by vertical orange edges.*

However, we need to address what to do when we encounter a split vertex on this path. Consider the vertex $v_{i,j}^{\ell,r}$ for some $j \in [k]$ and $r \in [n]$. If $v_{i,j}^{\ell,r}$ is not split, then we don't have to do anything. Otherwise, if $v_{i,j}^{\ell,r}$ is split then we add the edge $v_{i,j,LB}^{\ell,r} \leftarrow v_{i,j,TR}^{\ell,r}$ to P_i^ℓ.

Note that the orientation of G which orients all dotted edges leftwards, i.e., $LB \leftarrow TR$, contains each of the vertical canonical paths defined above. Observe that both the horizontal canonical paths and vertical canonical paths assign orientations to the dotted edges arising from splitting vertices. Hence, one needs to be careful because the splitting operation (see Figure 2) is designed to ensure that the existence of a horizontal canonical path implies that some vertical canonical path cannot exist (recall that we are allowed to orient each undirected edge in exactly one direction).

Definition 4. ***(realizable set of paths)*** *A set of directed paths $\mathscr{P}$ in a mixed graph G is* realizable *if there is a orientation G^* of G such that each path $P \in \mathscr{P}$ appears in G^*.*

Lemma 1. *The set of vertical canonical paths* $\{P_i^{\alpha_i} \ : \ i \in [k]\}$ *together with the set of horizontal canonical paths* $\{Q_j^{\beta_j} \ : \ j \in [k]\}$ *are realizable in G.*

Proof. Suppose to the contrary that this set of directed paths is not realizable in G. The only undirected edges which get oriented on horizontal canonical paths or vertical canonical paths are dotted edges which are created from the splitting operation. This implies that there is an undirected dotted edge, say $v_{i,j,LB}^{\alpha_i,\beta_j} - v_{i,j,TR}^{\alpha_i,\beta_j}$ which gets different orientations by the vertical canonical path $P_i^{\alpha_i}$ and the horizontal canonical path $Q_j^{\beta_j}$, respectively. This means that the black vertex $v_{i,j}^{\alpha_i,\beta_j}$ was split. However, by the property of the GRID TILING solution, we have that $(\alpha_i, \beta_j) \in S_{i,j}$ which contradicts the fact that $v_{i,j}^{\alpha_i,\beta_j}$ was split. $\square$

Observe that for each $j \in [k]$, the horizontal path $Q_j^{\beta_j}$ satisfies the two terminal pairs (b_j^1, c_j^n) and (b_j^n, c_j^1) for each $j \in [k]$ of Type V. Similarly, for each $i \in [k]$, the path $P_i^{\alpha_j}$ satisfies the two terminal pairs (f_j^1, g_j^n) and (f_j^n, g_j^1) of Type VI. Lemma 1 guarantees that these families of canonical vertical and horizontal paths can be realized by some orientation (note that the canonical paths only orient black edges, and not green edges whose orientation was already fixed at the start of this subsection) of G. This implies that the instance $(G, \mathcal{T})$ of STEINER ORIENTATION answers YES, and concludes this direction of the proof.

2.3 STEINER ORIENTATION has a solution ⇒ GRID TILING has a solution

Since the instance $(G, \mathcal{T})$ of STEINER ORIENTATION has a solution, let G^* be the orientation which satisfies all pairs from $\mathcal{T}$. Note that the set of vertices $B \cup D \cup F \cup H$ has no incoming edges. Similarly, the set of vertices $A \cup C \cup E \cup G$ has no outgoing edges.

Lemma 2. *No yellow edge can be on a path in* G^* *which satisfies any terminal pair of Type I or II*

Proof. Fix $j \in [k]$. We just prove the lemma for the terminal pair (b_j^n, a_j^1) since the proof for other terminal pairs is similar. Suppose there is a yellow edge on some path P satisfying the terminal pair (b_j^n, a_j^1). This yellow edge cannot be of Category I since D has no incoming edges, and hence we could not have reached D in the first place starting from b_j^n. However, this yellow edge also cannot be of Category II since H has no incoming edges and hence we could not have reached H in the first place starting from b_j^n. $\square$

The next lemma restricts the orientations of the undirected paths of green edges.

Lemma 3. *In the orientation* G^**, for each* $i \in [k]$ *we have that*

- *there exists an integer* $\lambda_i \in [n]$ *such that the paths* A_i, B_i *are oriented away from, and towards* λ_i*, respectively,*

– *there exists an integer $\mu_i \in [n]$ such that the paths C_i, D_i are oriented away from, and towards μ_i, respectively,*
– *there exists an integer $\delta_i \in [n]$ such that the paths E_i, F_i are oriented away from, and towards δ_i, respectively,*
– *there exists an integer $\varepsilon_i \in [n]$ such that the paths G_i, H_i are oriented away from, and towards ε_i, respectively.*

Proof. Fix $i \in [k]$. We just prove the lemma for the paths A_i, B_i since the proof for other cases is similar. By Lemma 2, we know that the paths satisfying the terminal pairs (b_i^n, a_i^1) and (b_i^1, a_i^n) cannot contain any yellow edges. Since the only non-yellow edges incoming to A are blue edges from B (and B has no incoming edges), it follows that the terminal pairs (b_i^n, a_i^1) and (b_i^1, a_i^n) of Type I are satisfied by edges from the graph $G^*[A_i \cup B_i]$. The path satisfying the terminal pair (b_i^n, a_i^1) has to travel downwards along B_i, use a blue edge and then finally travel downwards along A_i. Similarly, the path satisfying the terminal pair (b_i^1, a_i^n) has to travel upwards along B_i, use a blue edge and then finally travel upwards along A_i. Since we can only orient each green edge in exactly one direction, it follows that both these paths must use the same blue edge, i.e., there exists an integer $\lambda_i \in [n]$ such that the paths A_i, B_i are oriented away from, and towards λ_i, respectively. □

Lemma 4. *For each $i \in [k]$ and integers λ_i, μ_i, δ_i, ε_i as given by Lemma 3 we have that*

– $\lambda_i = \mu_i$,
– $\delta_i = \varepsilon_i$.

Proof. Fix $i \in [k]$. We just prove that $\lambda_i = \mu_i$ since the proof for the other case is similar. Consider the terminal pairs (d_i^n, a_i^1) and (d_i^1, a_i^n) of Type III. The only outgoing edges from D are to $A \cup C$. However, $A \cup C$ has not outgoing edges. Hence, the aforementioned terminal pairs are satisfied by edges from $G^*[D \cup A]$. By Lemma 3, we know that A_i is oriented away from λ_i and D_i is oriented towards μ_i. Hence, if $\mu_i > \lambda_i$ then the pair (d_i^n, a_i^1) is not satisfied, and if $\mu_i < \lambda_i$ then the pair (d_i^1, a_i^n) is not satisfied. Thus we have $\lambda_i = \mu_i$. □

Lemma 5. *No yellow edge can be on a path satisfying any terminal pair of Type V or VI .*

Proof. Fix $j \in [k]$. We just prove the lemma for the terminal pair (b_j^1, c_j^n) since the proof for other terminal pairs is similar. Suppose there is a yellow edge on some path P satisfying the terminal pair (b_j^1, c_j^n). This yellow edge cannot be of Category I since D has no incoming edges, and hence we could not have reached D in the first place starting from b_j^1. However, this yellow edge also cannot be of Category II since H has no incoming edges and hence we could not have reached H in the first place starting from b_j^1. □

Lemma 6. *For each $1 \leq i, j \leq k$, we have that*

– *any path which satisfies the terminal pair (b_j^1, c_j^n) **must** contain the horizontal canonical path $Q_j^{\lambda_j}$,*

– *any path which satisfies the terminal pair (f_i^n, g_i^1) **must** contain the vertical canonical path $P_i^{\delta_i}$.*

Proof. Fix $j \in [k]$. Consider the terminal pair (b_j^1, c_j^n), and let P be any path satisfying it. By Lemma 5, we know that P cannot have any yellow edges. By Lemma 3 and Lemma 4, we know that B_j, C_j are oriented towards, and away from λ_j, respectively. We claim that the first edge on P which leaves B_j is $b_j^{\lambda_j} \to v_{1,j}^{1,\lambda_j}$. Clearly, P cannot have any edge from B_j to A_j, since A_j has no outgoing edges. Hence, the path P is of the following type: the vertical upwards path $b_j^1 \to b_j^2 \to \ldots b_j^\tau$ followed by the blue edge $b_j^\tau \to v_{1,j}^{1,\tau}$. Since B_j is oriented towards λ_j it follows that $\lambda_j \geq \tau$. If $\lambda_j > \tau$ then by orientation of the black grid edges and orange edges (note that the splitting doesn't really change the rows/colums level) it follows that P reaches C_j at a vertex c_j^ψ where $\lambda_j > \tau \geq \psi$. However, C_j is oriented away from λ_j which contradicts that P is a path from b_j^1 to c_j^n. Hence, we have that $\lambda_j = \tau = \psi$. Therefore, P contains a subpath which starts at $b_j^{\lambda_j}$ and ends at $c_j^{\lambda_j}$ and all edges of this subpath (except the first and last blue edges) are contained in the graph $G^*\left[\bigcup_{i=1}^k V(G_{i,j})\right]$, i.e., P contains the canonical horizontal path $Q_j^{\lambda_j}$.

The proof of the second part of the lemma is similar, and we omit the details here. □

Lemma 7. *The instance $(k, n, \{S_{i,j}\}_{1 \leq i,j \leq k})$ of* Grid Tiling *has a solution.*

Proof. We show that $(\delta_i, \lambda_j) \in S_{i,j}$ for each $1 \leq i, j \leq k$. This will imply that Grid Tiling has a solution.

Fix any $1 \leq i, j \leq k$. By Lemma 6, we know that the orientation G^* must contain the horizontal canonical path $Q_j^{\lambda_j}$ (to satisfy the pair (b_j^1, c_j^n)) and also the vertical canonical path $P_i^{\delta_i}$ (to satisfy the pair (f_i^n, g_i^1)). We now claim that the vertex $v_{i,j}^{\delta_i,\lambda_j}$ cannot be split: suppose to the contrary that it is split. By Definition 3, the path $P_i^{\delta_i}$ orients the edge $v_{i,j,LB}^{\delta_i,\lambda_j} - v_{i,j,TR}^{\delta_i,\lambda_j}$ as $v_{i,j,LB}^{\delta_i,\lambda_j} \leftarrow v_{i,j,TR}^{\delta_i,\lambda_j}$. However, by Definition 2, the path $Q_j^{\lambda_j}$ orients the edge $v_{i,j,LB}^{\delta_i,\lambda_j} - v_{i,j,TR}^{\delta_i,\lambda_j}$ as $v_{i,j,LB}^{\delta_i,\lambda_j} \to v_{i,j,TR}^{\delta_i,\lambda_j}$, which is a contradiction. Hence, the vertex $v_{i,j}^{\delta_i,\lambda_j}$ is not split, i.e., $(\delta_i, \lambda_j) \in S_{i,j}$ for each $1 \leq i, j \leq k$. □

2.4 Obtaining the $f(k) \cdot n^{o(k)}$ lower bound

It is easy to see that the graph G has $O(n^2k^2)$ vertices and can be constructed in poly$(n+k)$ time. Combining the two directions from Subsections 2.2 and 2.3, we get a paramterized reduction from Grid Tiling to Steiner Orientation. Hence, the W[1]-hardness of Steiner Orientation follows from the W[1]-hardness of Grid Tiling [11]. Chen et al. [2] showed that, for any function f, the existence of an $f(k) \cdot n^{o(k)}$ algorithm for k-Clique violates ETH. There is a simple reduction (see Theorem 14.28 from [6]) from k-Clique to $k \times k$ Grid Tiling implying the same runtime lower bound for the latter problem. Our reduction transforms the problem of $k \times k$ Grid

TILING into an instance of STEINER ORIENTATION with $O(k)$ demand pairs. Composing the two reductions, we obtain that under ETH there is no $f(k) \cdot n^{o(k)}$ time algorithm for STEINER ORIENTATION. Recall from Remark 1 that the graph G constructed in the STEINER ORIENTATION instance has genus 1, and hence the $f(k) \cdot n^{o(k)}$ lower bound holds for genus 1 graphs too. This concludes the proof of Theorem 1.

References

1. E. M. Arkin and R. Hassin. A note on orientations of mixed graphs. *Discrete Applied Mathematics*, 116(3):271–278, 2002.
2. J. Chen, X. Huang, I. A. Kanj, and G. Xia. Strong Computational Lower Bounds via Parameterized Complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.
3. R. Chitnis, H. Esfandiari, M. T. Hajiaghayi, R. Khandekar, G. Kortsarz, and S. Seddighin. A Tight Algorithm for Strongly Connected Steiner Subgraph on Two Terminals with Demands. *Algorithmica*, 77(4):1216–1239, 2017.
4. R. Chitnis, A. E. Feldmann, and P. Manurangsi. Parameterized Approximation Algorithms for Directed Steiner Network Problems. *CoRR*, abs/1707.06499, 2017.
5. R. H. Chitnis, M. Hajiaghayi, and D. Marx. Tight Bounds for Planar Strongly Connected Steiner Subgraph with Fixed Number of Terminals (and Extensions). In *SODA*, pages 1782–1801, 2014.
6. M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
7. M. Cygan, G. Kortsarz, and Z. Nutov. Steiner forest orientation problems. *SIAM J. Discrete Math.*, 27(3):1503–1513, 2013.
8. R. Hassin and N. Megiddo. On orientations and shortest paths. *Linear Algebra and its Applications*, 114:589 – 602, 1989. Special Issue Dedicated to Alan J. Hoffman.
9. R. Impagliazzo and R. Paturi. On the Complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
10. R. Impagliazzo, R. Paturi, and F. Zane. Which Problems Have Strongly Exponential Complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001.
11. D. Marx. On the Optimality of Planar and Geometric Approximation Schemes. In *FOCS*, pages 338–348, 2007.
12. D. Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010.
13. D. Marx. A Tight Lower Bound for Planar Multiway Cut with Fixed Number of Terminals. In *ICALP (1)*, pages 677–688, 2012.
14. D. Marx and M. Pilipczuk. Everything you always wanted to know about the parameterized complexity of Subgraph Isomorphism (but were afraid to ask). In *STACS*, pages 542–553, 2014.
15. D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using voronoi diagrams. In *ESA*, pages 865–877, 2015.
16. M. Pilipczuk and M. Wahlström. Directed multicut is W[1]-hard, even for four terminal pairs. *CoRR*, abs/1507.02178, 2015.
17. M. Pilipczuk and M. Wahlström. Directed multicut is $W[1]$-hard, even for four terminal pairs. In *SODA*, pages 1167–1178, 2016.