Planar Steiner Orientation is NP-complete

Moritz Beck — **Julius-Maximilians-Universität Würzburg, Germany**
Johannes Blum — **Julius-Maximilians-Universität Würzburg, Germany**
Myroslav Kryven — **Julius-Maximilians-Universität Würzburg, Germany**
Andre Löffler — **Julius-Maximilians-Universität Würzburg, Germany**
Johannes Zink — **Julius-Maximilians-Universität Würzburg, Germany**

Abstract

Many applications in graph theory are motivated by routing or flow problems. Among these problems is Steiner Orientation: given a mixed graph G (having directed and undirected edges) and a set T of k terminal pairs in G, is there an orientation of the undirected edges in G such that there is a directed path for every terminal pair in T? This problem was shown to be *NP*-complete by Arkin and Hassin [1] and later $W[1]$-hard by Pilipczuk and Wahlström [7], parametrized by k. On the other hand, there is an *XP* algorithm by Cygan et al. [3] and a polynomial time algorithm for graphs without directed edges by Hassin and Megiddo [5]. Chitnis and Feldmann [2] showed $W[1]$-hardness of the problem for graphs of genus 1.

We consider a further restriction to planar graphs and show *NP*-completeness.

1 Introduction

Consider the following routing problem on mixed graphs:

	Steiner Orientation
Input:	Mixed graph $G = (V, E \cup A)$ with undirected edges E and directed arcs A, set $T \subseteq V \times V$ of k terminal pairs
Output:	Orientation of all edges in E, such that for every $(s,t) \in T$ there is an s-t-path in G

First notice that there is a polynomial time algorithm by Hassin and Megiddo [5] for the case where G has no directed edges ($A = \emptyset$). On general graphs, this problem is in fact *NP*-complete, shown by Arkin and Hassin [1], also providing an efficient algorithm for the special case of $k = 2$. Generalizing this result to $k \geq 1$, Cygan et al. [3] gave a $n^{O(k)}$ time algorithm. This means that Steiner Orientation is in *XP*, parametrized by the number of terminal pairs. This raised the question if Steiner Orientation is *fixed-parameter tractable*: is there an algorithm with runtime $f(k) \cdot n^{O(1)}$ for some computable function f only dependent on k? Pilipczuk and Wahlström [7] showed Steiner Orientation to be $W[1]$-hard in k, disproving the existance of such an algorithm under common assumptions. Considering k-SAT, Impagliazzo and Paturi [6] introduced the *Exponential Time Hypothesis* (ETH): the classic SAT-problem parametrized by the number of variables per clause does not have a subexponential algorithm. Assuming ETH—a common assumption when parametrized problems are considered—Pilipczuk and Wahlström [7] could also show that there is no $f(k) \cdot n^{o(k/\log k)}$ time algorithm for any computable f, showing that the *XP* algorithm by Cygan et al. [3] is almost optimal. Even when restricting to graphs of genus 1, Chitnis and Feldmann [2] showed that the problem remains $W[1]$-hard and the *XP* algorithm still is almost optimal. All hardness-proofs provided utilize non-planar instances. This leaves open the following question: What is the computational complexity of Steiner Orientation on planar graphs?

In this work, we consider the Planar Steiner Orientation problem where G is a planar graph. As a first result on computational complexity, we show the following:

Theorem 1. Planar Steiner Orientation *is NP-complete.*

2 Hardness Proof

To prove Theorem 1, we give a reduction from Planar Monotone 3-SAT, introduced by de Berg and Khosravi [4] and known to be *NP*-complete. We use different gadgets for variables, clauses and edges. These are stitched together at shared undirected edges. Given a planar monotone 3-SAT formula F, we use these gadgets to create an instance of Planar Steiner Orientation resembling the incidence graph of F with $|T|$ polynomial in the size of F. Without loss of generality we assume that every variable of F occurs both negated and unnegated.

Figure 1 (a) shows a *flip gadget*, a building block used in other gadgets. It contains two terminal pairs (s_1, t_1) and (s_2, t_2) and two undirected (red) edges. Connecting both pairs will result in opposing directions for the two undirected edges.

For every variable x in F, we have a *variable gadget* (Figure 1 (b)). It mimics the flip gadget, providing an undirected edge e_C^x for every positive/negative clause C containing x above/below the terminal pairs respectively. We say that the gadget is (false) true if the undirected edges are oriented (counter-)clockwise. No other orientation allows connecting both pairs.

For every clause C, we have a *clause gadget* (Figure 1 (c)). It contains a terminal pair (s, t) and has an undirected edge $\overline{e}_C^w$ for each variable w it contains. The undirected edge $\overline{e}_C^y$ in the middle is flipped to get a consistent orientation for variables set to true. The edges f and g are synchronized by two flip gadgets to ensure that at most one of them is used to connect (s, t). For clauses with only two variables we simply replace the edge $\tilde{e}_C^y$ with an arc from left to right and omit the attached flip gadget (see Figure 2 (a), clause $(X \vee Y)$). It is easy to see that this way no new possibilities for s-t-paths are created, keeping the gadget valid.

We use two stacked flip gadgets as *edge gadgets* resembling the variable-clause-incidences. By reversing the direction twice, we synchronize the two red edges e_C^x and $\overline{e}_C^x$ for all x and C.

An important property of our construction is that all the gadgets that we use are *self-contained*, which means that for each gadget any simple path connecting an (s, t)-pair of the gadget stays inside the gadget. We state this simple observation, which we use to prove the following lemma:

Observation 1. *Every source s has indegree zero and every target t has outdegree zero.*

Lemma 1. *In our construction each clause, edge, and variable gadget is self-contained.*

Proof. **Clause gadgets.** Assume there is a simple path connecting a terminal pair of a clause gadget, which is not fully contained within the gadget. Then there must be an edge that leaves the clause gadget and due to the structure, this edge must be part of an edge gadget. But all edges leaving a clause gadget and entering an edge gadget end in some target terminal, so by Observation 1 the path cannot re-enter the clause gadget.

Edge gadgets. Consider a simple path that leaves an edge gadget. If the leaving edge is part of a clause gadget, the path leads to a target terminal within the clause gadget or within another edge gadget, from where it cannot re-enter the original edge gadget. The case where the leaving edge is part of a variable gadget is similar.

Variable gadgets. Consider a simple path that leaves a variable gadget. Then the leaving edge is part of an edge gadget and leads to a target terminal, so the path cannot re-enter the variable gadget. □

Using this, we obtain the following lemmas regarding our gadgets.

Lemma 2. *All terminal pairs of a clause gadget corresponding to a clause C can be connected if and only if at least one of the edges in $\{\overline{e}_C^x, \overline{e}_C^y, \overline{e}_C^z\}$ is directed to the right.*

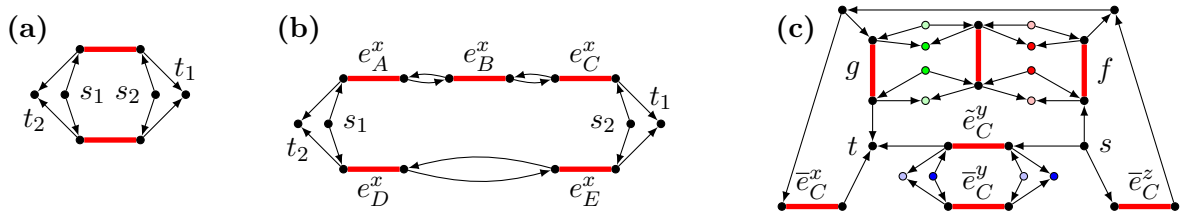


Figure 1: **(a)** The flip gadget, used to construct edge gadgets; **(b)** a variable gadget with three positive and two negative occurrences; **(c)** a clause gadget (unlabeled (s,t)-pairs color-coded).

Proof. According to Lemma 1 it suffices to consider only paths within the clause gadget as there is a path connecting a terminal pair if and only if there is a simple path connecting this terminal pair. The terminal pairs of the flip gadgets can be connected within the flip gadgets. They cannot be connected otherwise within the clause gadget since they would have to pass through a source or target vertex then, which contradicts Observation 1. The edges f and g are both directed upwards or downwards because of the two flip gadgets between them. Hence it suffices to show the equivalence for the pair (s,t).

"$\Leftarrow$": Case 1: If $\overline{e}_C^x$ is directed to the right, orient the edge f away from s. Case 2: If $\overline{e}_C^y$ is directed to the right, $\tilde{e}_C^y$ is directed to the left. Case 3: If $\overline{e}_C^z$ is directed to the right, orient the edge g pointing to t. In each of these cases s is connected to t.

"$\Rightarrow$": By contraposition. As we move away from s, we can neither use the edge $\tilde{e}_C^y$ nor $\overline{e}_C^z$. Thus we have to use f which means it points away from s. To come to t we have to use one of the edges $\overline{e}_C^x$ or g. This is impossible. □

The next two lemmas follow immediately from the gadget structure and Lemma 1.

Lemma 3. *The only way to connect all terminal pairs of a variable gadget is to orient all edges clockwise or counterclockwise.*

Lemma 4. *The only way to connect all terminal pairs of an edge gadget is to orient the two outer edges e_C^x and $\overline{e}_C^x$ in the same direction.*

We can observe that there is always an orientation such that the terminal pairs of all variable and edge gadgets can be connected. Provided that all terminal pairs of all variable and edge gadgets are connected, the terminal pairs of all clause gadgets can be connected if and only if the formula is satisfiable. Thus, the Planar Steiner Orientation instance has a solution if and only if the corresponding Planar Monotone 3-SAT formula is satisfiable. This proves Theorem 1.

Full Example We want to provide a small but complete example. In Figure 2 we give the incidence graph and the Planar Steiner Orientation instance created using the gadgets introduced above corresponding to: $F = (X \vee Y) \wedge (\neg X \vee \neg Z \vee \neg W) \wedge (Y \vee Z \vee W) \wedge (\neg X \vee \neg Y \vee \neg Z)$.

3 Conclusion

By a polynomial-time reduction from Planar Monotone 3-SAT we have shown that Planar Steiner Orientation is *NP*-hard. Clearly, it also is in *NP*, which makes it *NP*-complete.

Future work involves proving $W[1]$-hardness or looking for approximation algorithms, connecting as many pairs as possible. Graph classes with other restrictions could also be considered.

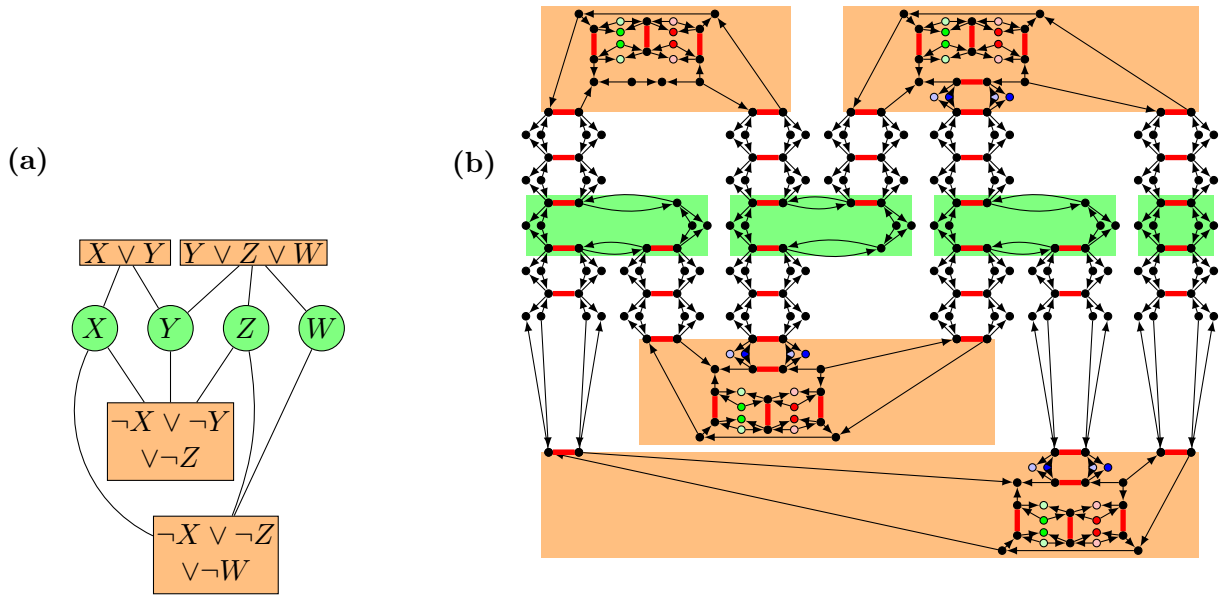


Figure 2: A full example showing the reduction from Planar Monotone 3-SAT to Planar Steiner Orientation: (a) incidence graph for formula F; (b) corresponding reduction instance. Variables and variable gadgets are highlighted in green, clauses and clause gadgets in orange.

Acknowledgements We thank Andreas Feldmann for pointing us to this problem and we also thank the workshop HOMONOLO 2017 for the fruitful discussions.

References

[1] Esther M. Arkin and Refael Hassin. A note on orientations of mixed graphs. *Discrete Applied Mathematics*, 116(3):271–278, 2002.

[2] Rajesh Chitnis and Andreas Emil Feldmann. A Tight Lower Bound for Steiner Orientation. In *Proceedings of the 13th Intl. Computer Science Symposium in Russia*, 2018. (to appear).

[3] Marek Cygan, Guy Kortsarz, and Zeev Nutov. Steiner Forest Orientation Problems. *SIAM Journal on Discrete Mathematics*, 27(3):1503–1513, 2013.

[4] Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal of Computational Geometry & Applications*, 22(3):187–206, 2012.

[5] Refael Hassin and Nimrod Megiddo. On orientations and shortest paths. *Linear Algebra and its Applications*, 114:589–602, 1989.

[6] Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

[7] Marcin Pilipczuk and Magnus Wahlström. Directed multicut is W[1]-hard, even for four terminal pairs. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1167–1178, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2884435.2884516.