Extending the Kernel for Planar Steiner Tree to the Number of Steiner Vertices

Ondřj Suchý*

Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Czech Republic
ondrej.suchy@fit.cvut.cz

Abstract

In the Steiner Tree problem one is given an undirected graph, a subset T of its vertices, and an integer k and the question is whether there is a connected subgraph of the given graph containing all the vertices of T and at most k other vertices. The vertices in the subset T are called terminals and the other vertices are called Steiner vertices. Recently, Pilipczuk, Pilipczuk, Sankowski, and van Leeuwen [FOCS 2014] gave a polynomial kernel for Steiner Tree in planar graphs, when parameterized by $|T| + k$, the total number of vertices in the constructed subgraph.

In this paper we present several polynomial time applicable reduction rules for Planar Steiner Tree. In an instance reduced with respect to the presented reduction rules, the number of terminals $|T|$ is at most quadratic in the number of other vertices k in the subgraph. Hence, using and improving the result of Pilipczuk et al., we give a polynomial kernel for Steiner Tree in planar graphs for the parameterization by the number k of Steiner vertices in the solution.

1998 ACM Subject Classification E.4 Coding and Information Theory, F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases Steiner Tree, polynomial kernel, planar graphs, polynomial-time preprocessing, network sparsification

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.151

1 Introduction

The Steiner problem is a classical problem of theoretical computer science and a fundamental problem of network design. Most generally it can be formulated as follows: Given a set of interesting objects in some environment or a network, determine the most efficient way to connect them. The study of Steiner Tree in graphs goes back to Hakimi [24] (the problem was also independently formulated by Levin [32]), who showed that Clique can be reduced to Steiner Tree (the theory of NP-hardness was not known yet). Applications can be found in VLSI routing [29], network routing in general [31], phylogenetic tree reconstruction [27] and other areas. It was also the topic of the 11th DIMACS Implementation Challenge. We refer the reader to one of many books devoted to Steiner Tree for further applications [10, 20, 37].

Our focus in this paper is on the case of planar graphs. Hence, we consider the following formulation of the problem:

Planar Steiner Tree
Input: A planar graph $G = (V, E)$, a set $T \subseteq V$, and an integer k.
Question: Is there a set $S \subseteq V \setminus T$ of size $|S| \leq k$ such that $G[T \cup S]$ is connected?

* The research was partially supported by the grant 14-13017P of the Czech Science Foundation.

We denote n the number of vertices of the graph G and m the number of its edges. We call the vertices in T *terminals* and the vertices in $V \setminus T$ *non-terminals* or *Steiner vertices*. We call a set $S \subseteq V \setminus T$ a solution (for the instance (G, T, k)), if $|S| \leq k$ and for every pair of vertices $x, y \in T$ the vertices x and y are in the same connected component of $G[S \cup T]$. Note that then all vertices of T are in the same connected component of $G[S \cup T]$ and we can remove all other connected components from the graph and get a set asked by the problem definition.

The problem is often formulated so that one is to find a connected subgraph with the minimum number of edges that contains the set of terminals. The subgraph attaining the minimum must be a tree, called *optimal Steiner tree*. It is not hard to see that there is such a connected subgraph with $|T| + k - 1$ edges if and only if there is a set of vertices forming a solution to our formulation (with k Steiner points).

Our Contribution

The problem is NP-hard [23] and remains so even in very restricted planar cases [22]. In order to better understand the complexity of the problem we focus on the parameterized analysis of the problem. The problem was recently studied with respect to the parameter "the number of edges in an optimal Steiner tree" or equivalently $|T| + k$ by Pilipczuk, Pilipczuk, Sankowski, and van Leeuwen [35, 36], who obtained a subexponential algorithm [35] and a polynomial kernel [36] for the problem with respect to this parameterization. In particular they proved the following proposition.

▶ **Proposition 1** (Pilipczuk et al. [36]). *Given a* Planar Steiner Tree *instance* (G, T)*, one can in* $O(k_{OPT}^{142} n)$ *time find a set* $F \subseteq E(G)$ *of* $O(k_{OPT}^{142})$ *edges that contains an optimal Steiner tree connecting* T *in* G*, where* k_{OPT} *is the total number of edges of an optimal Steiner tree.*

In this paper we focus on the parameterization by the number k of Steiner vertices in the solution. By folklore result Planar Steiner Tree is known to be fixed parameter tractable with respect to this parameter (see also [28]), however it was not known whether there is a polynomial kernel. We resolve this question as follows: We present several polynomial time reduction rules and show that if the rules are exhaustively applied the number of terminals is at most quadratic in k. Then the number of edges of an optimal Steiner tree of that instance is also at most quadratic in k and we can use the algorithm of Proposition 1 to obtain a polynomial kernel.

This improves the result of Pilipczuk et al. qualitatively, since we give a polynomial kernel with respect to a parameter that is always smaller and can be arbitrarily smaller than the parameter they use. Moreover, it also improves it quantitatively, as our rules never increase the number of edges in an optimal Steiner tree, and, hence, the kernel obtained by first running our rules is always at most as big as the one obtained by starting directly with the algorithm of Proposition 1.

Related Work

As we already mentioned the problem is NP-complete even in very restricted cases of planar graphs [22]. It is also well studied from the approximation perspective. It can be approximated to within a factor $O(\log n)$, but it cannot be approximated within a factor $(1 - \varepsilon)(\log |T|)$ unless NP $\subseteq$ DTIME$[n^{\text{polylog}(n)}]$ [30]. Furthermore, the edge weighted variant admits a constant factor approximation [9], while it is APX-complete even on complete graphs with weights 1 and 2 [2].

Many studies considered the planar variant of the problem from the approximation perspective. The edge weighted variant admits EPTAS on planar graphs [8] as well as on bounded genus graphs [7]. The number of papers on approximation of some variant of Steiner type problem is enormous and rapidly growing. Therefore we refer the reader to the online compendium [25] for the current approximation state of various Steiner type problems.

Turning to exact exponential algorithms, there is a simple folklore algorithm running in $O(2^{\frac{2}{3}n}n^{O(1)}) = O(1.6181^n)$ time. This was improved to $O(1.59^n)$ for the weighted case and $O(1.36^n)$ for the cardinality case by Fomin et al. [17]. Note that all of the mentioned algorithms use polynomial space.

Concerning parameterized complexity, on general graphs, Steiner Tree is known to be W[2]-hard with respect to the standard parameterization k [14]. Moreover, by result of Patrascu and Williams [34], there are no $l \geq 3$, $\varepsilon > 0$, and an algorithm that would solve Steiner Tree on instance (G, T, k) with $k = l$ in time $O(n^{l-\varepsilon})$, unless the Strong Exponential Time Hypothesis (SETH) fails.

However, for Steiner Tree much more often the parameterization by the number of terminals $|T|$ is used. There is a nice long history of improving FPT-algorithms with respect to this parameterization started by the $O(3^{|T|} \cdot n + 2^{|T|} \cdot n^2 + n(n \log n + m))$-time algorithm of Dreyfus and Wagner [15] (independently found by Levin [32]). This algorithm, as well as its later improvements [16, 21, 4] subsequently approaching the $O^*(2^{|T|})$ running time, use exponential space.

Polynomial space FPT-algorithms appeared only recently. The one by Nederlof [33] applies to the cardinality variant and the variant where the weights are bounded by a constant and achieves $O^*(2^{|T|})$ time. The running time of $O^*(2^{|T|})$ is believed to be optimal [11]. The algorithm by Fomin et al. [18] achieves running time $O(7.97^{|T|} \cdot n^4 \cdot \log W)$ for edge weights in $\{1, \ldots, W\}$.

On general graphs the problem does not admit polynomial kernel even with respect to $k + |T|$, unless coNP $\subseteq$ NP/poly. This can be easily proved using the framework of Bodlaender et al. [6], a less direct approach can be found in [13]. Note also that since the problem is on general graphs FPT with respect to $|T|$ and W[2]-hard with respect to k, no reduction of type presented in our paper can exist for general graphs, unless W[2]=FPT.

By way of contrast, much less is known about the parameterized complexity of Planar Steiner Tree. In fact, for the parameterization by $|T|$, no results are known that would improve those from general graphs. On the other hand, for the parameterization by k, the problem is known to be FPT by a folklore result based on Steiner Tree being FPT with respect to the treewidth of the graph G [12, 5, 19]. This was improved by Jones et al. [28], who presented an algorithm in $O^*(3^{dk+o(dk)})$ that applies to d-degenerate graphs.

However, recently there was a significant progress for planar graphs with respect to the combined parameter $k + |T|$. First Pilipczuk et al. [35] showed that there is a subexponential algorithm for Planar Steiner Tree with respect to this parameterization running in $O(2^{O(((k+|T|)\log(k+|T|))^{2/3})}n)$ time. Later the same group of authors improved this to $O(2^{O(\sqrt{(k+|T|)\log(k+|T|)})}n)$ time and gave the kernel as presented in Proposition 1 [36].

Organization of the paper

Our algorithm is described in Section 2. In particular Subsection 2.1 contains the reduction rules that apply to all graphs, Subsection 2.2 collects the rules that only hold in planar graphs and Subsection 2.3 contains the main theorem and the analysis of the running time. We conclude the paper by giving some outlooks for future research in Section 3.

Due to space constraints, several proofs had to be deferred to the full version of the paper.

2 Algorithm

Our algorithm consist of several reduction rules, which simplify the instance. The algorithm applies these rules exhaustively to obtain an instance which is *reduced* with respect to them, that is, an instance to which none of the rules applies any more. We give the rules in a specific order and always prefer to apply a rule that was given earlier to a rule given later. In other words, before we apply some rule, we assume that the instance is reduced with respect to all previous rules.

For each of the rules we immediately prove its *correctness*, that is, the instance produced by the rule is a yes-instance if and only if the original instance was (the instances are *equivalent*). However we defer the analysis of the running times to find applications and to apply the rules until all the rules are presented.

We present the rules and some auxiliary lemmata that apply to general graphs in Subsection 2.1, rules for planar graphs in Subsection 2.2 and the running time and the main theorem in Subsection 2.3.

2.1 General Reduction Rules

Here we give the rules and lemmata that would apply to general graphs, however also preserve planarity of the instance. We start by a rule that summarizes the obvious trivial constraints on solvability of the instance. Its correctness is immediate.

▶ **Reduction Rule 1.**
(a) If $k \geq 0$ and $\emptyset$ is a solution, then answer YES.
(b) If $k < 0$, then answer NO.
(c) If $k = 0$ and $\emptyset$ is not a solution, then answer NO.
(d) If for some $x, y \in T$ there is no path between x and y in G, then answer NO.

We continue by a well known rule for Steiner Tree.

▶ **Reduction Rule 2** (Folklore)**.** If there are two adjacent terminals x and y, then contract the edge $\{x, y\}$. I.e., we continue with the instance (G', T', k), where $G' = (V', E')$, $V' = (V \setminus \{x, y\}) \cup \{w\}; w \notin V$, $E' = (E \setminus \{e \mid e \in E, e \cap \{x, y\} \neq \emptyset\}) \cup \{(e \setminus \{x, y\}) \cup \{w\} \mid e \in E, |e \cap \{x, y\}| = 1\})$, and $T' = (T \setminus \{x, y\}) \cup \{w\}$.

Although the rule is well known, we prove its correctness for completeness.

▶ **Lemma 2** (★[1])**.** *Reduction Rule 2 is correct.*

The following lemma pinpoints the property of solutions that is crucial for our considerations in the rest of the paper. We use the following notion: a Steiner vertex v *dominates* a terminal x if v and x are adjacent in G.

▶ **Lemma 3.** *Let (G, T, k) be an instance where Reduction Rules 1 and 2 have been exhaustively applied, S a solution of the instance, and x a vertex in T. Then there is a vertex v in S such that v dominates x.*

Proof. Since the instance is reduced with respect to Reduction Rule 1, we know that there are at least two terminals, as otherwise $\emptyset$ would be a solution. Let y be a terminal different from x. Since S is a solution, there is a path $p_1, p_2, \ldots, p_q$ in $G[S \cup T]$ such that $p_1 = x$ and

[1] Proofs of lemmata marked with (★) were deferred to the full version of the paper.

$p_q = y$. Let $v = p_2$. If v is in T, then x and v are two adjacent terminals contradicting the instance being reduced with respect to Reduction Rule 2. Hence v is in S and dominates x as claimed. ◀

We will use the following lemma in our proofs to show that taking some vertex to a solution can be modeled by a suitable modification of the instance and this preserves yes-instances.

▶ **Lemma 4.** *Let (G, T, k) be an instance, S a solution for (G, T, k), and v a vertex in S such that there is a terminal in the same connected component of $G[T \cup S]$ as v. Let us denote $T' = T \cup \{v\}$ and $k' = k - 1$. Then (G, T', k') is a yes-instance.*

The condition on v being in the same connected component of $G[T \cup S]$ as some terminal might seem a little strange and it would be more natural to just assume S to be minimal. However, we prefer to formulate the lemma this way, as it makes it easier to use.

Proof. Let us denote $S' = S \setminus \{v\}$. Then $|S'| \leq k'$ and $T \cup S$ equals $T' \cup S'$. Thus for every pair of vertices y, z from T the terminals y and z are in the same connected component of $G[T' \cup S'] = G[T \cup S]$. Moreover, v is in the same connected component of $G[T' \cup S'] = G[T \cup S]$ as at least one other terminal and, thus, all other terminals, by assumption. Hence S' is a solution for (G, T', k'), finishing the proof. ◀

The following lemma is complementary to the previous one and shows that the operation preserves no-instances.

▶ **Lemma 5.** *Suppose (G, T, k) is an instance, $v \in V \setminus T$, and $x \in T$. Let us denote $T' = T \cup \{v\}$ and $k' = k - 1$. If (G, T', k') is a yes-instance, then (G, T, k) is a yes-instance.*

Proof. Let S' be a solution for (G, T', k') and let us denote $S = S' \cup \{v\}$. Then $|S| \leq k$ and again $T \cup S$ equals $T' \cup S'$ and, thus, $G[T \cup S] = G[T' \cup S']$. Since $T \subseteq T'$, S is a solution for (G, T, k), finishing the proof. ◀

The following rule shows that (false) twins among the terminals are superfluous. Surprisingly, we were not able to find the use of such a rule for Steiner Tree in literature.

▶ **Reduction Rule 3.** If there are x and y in T such that $N(x) = N(y)$, then remove x from G. I.e., we continue with instance (G', T', k), where $G' = G \setminus x$ and $T' = T \setminus \{x\}$.

▶ **Lemma 6 (★).** *Reduction Rule 3 is correct.*

2.2 Plane Specific Reduction Rules

In this subsection we present rules that rely on the graph being planar. To ease the presentation it is better to fix an embedding of the graph. Hence, for the rest of the paper we assume, that the given graph G is plane, i.e., it has a fixed planar embedding. Since a planar embedding can be found in linear time [26], this assumption is not restrictive. In fact we often consider the sphere embedding, as we mostly do not distinguish the outer face.

Consider two non-terminals u and v with at least two common terminal neighbors. Denote the set of common terminal neighbors Q and $q = |Q|$. The embedding of the edges connecting the vertices u and v to the vertices of Q cuts the surface of the sphere into q connected areas. Let A be any of these areas. If x and y are the two vertices of Q incident to the area A, then we say that u, x, v, and y form an *eye* and the cycle u, v, x, y forms its *boundary*. Moreover, we say that a vertex is *inside* the eye u, x, v, y, if it is embedded inside the area A. We say that a vertex is *outside* the eye if it is neither inside nor on the boundary of it. Note that a terminal z inside the eye u, x, v, y can be connected to u or v, but not to both of them by the definition of an eye.

Our first planar rule applies to eyes where one of the Steiner points on the boundary dominates all the terminals inside the eye.

▶ **Reduction Rule 4.** Suppose u, x, v, and y form an eye, such that $u, v \in V \setminus T$, and $x, y \in T$. Further suppose that there is a terminal $z \in T$ inside the eye. If every terminal inside the eye is a neighbor of v, then add v to T and reduce k by one.

The intuition behind the proof of correctness is that taking v into the solution is always at least as good as taking any vertex inside the eye.

▶ **Lemma 7 (★).** *Reduction Rule 4 is correct.*

The next rule allows to make an eye which does not contain terminals completely empty. We require the eye to be non-empty before the application of the rule so that the application actually changes the graph.

▶ **Reduction Rule 5.** Suppose u, x, v, and y form an eye, such that $u, v \in V \setminus T$ and $x, y \in T$ and suppose that there is a vertex $w \in V \setminus T$, but no terminal, inside the eye. Then remove every vertex w inside the eye from G.

Intuitively, if any vertex inside the eye is to be in the solution, then we can take u (or v) and detour any path around the eye.

▶ **Lemma 8 (★).** *Reduction Rule 5 is correct.*

The following variant of the so-called "high degree rule" forms the crux of our algorithm.

▶ **Reduction Rule 6.** If there is a vertex $u \in V \setminus T$ which dominates more than $5k$ terminals, then add u to T and reduce k by one. I.e., continue with instance (G, T', k'), where $k' = k - 1$ and $T' = T \cup \{u\}$.

▶ **Lemma 9.** *Reduction Rule 6 is correct.*

Proof. By Lemma 5, if the resulting instance is a yes-instance, then so is the original one. Now for the other direction, assume that there is a set $S \subseteq V \setminus T$ that is a solution for (G, T, k). If S contains u, then the resulting instance is a yes-instance by Lemma 4. Hence, let us assume for the rest of the proof that $u \notin S$. We show that this leads to a contradiction with the instance being reduced with respect to the previous rules.

Consider the terminals in $N(u)$. Each of them is dominated by a vertex of S by Lemma 3. Let B be the set of vertices in S dominating at least 2 vertices in $N(u) \cap T$. Each vertex in B forms at least 2 eyes together with u. We want to show that there must be a vertex in B for which many of these eyes are empty. To this end, let us fix a face of the embedding of G as the outer one and call an eye *outer* if it contains the outer face and *internal* otherwise. We will use the following auxiliary claim about the eyes between u and vertices in B.

▶ **Claim 1 (★).** *There are no two internal eyes u, x, b, y and u, x', c, y' with $b \neq c$ such that the union of their boundaries cuts the plane into 4 regions.*

Let us define the following relation on vertices of S. For $a, b \in S$ we write $a < b$ if and only if b is in B and a is inside some internal eye formed by u and b. We show that this relation can be used to order the vertices of S.

▶ **Claim 2 (★).** *The relation $<$ is a strict partial order.*

Let us now define another relation $\prec$ as follows. We write $a \prec b$ if and only if $a < b$ and there is no c such that $a < c$ and $c < b$. This relation is the cover relation of the strict partial order $<$. We need the following property of that relation.

▶ **Claim 3** (★). *For every a there is at most one b such that $a \prec b$.*

For a vertex b in B we call the set of vertices a such that $a \prec b$ the *support* of b and denote $\text{supp}(b)$. The support gives each vertex of B a budget for nonempty eyes in the following sense.

▶ **Claim 4** (★). *Let b be a vertex in B and consider an eye A between u and b containing a vertex of S. Then there is a vertex a inside A such that $a \prec b$ and, hence, a is in $supp(b)$.*

Next we show that, since the budget is limited, there are vertices which dominate more vertices than what their budget allows them.

▶ **Claim 5.** *There is a vertex b of B which dominates more than $2|supp(b)| + 3$ vertices of $T \cap N(u)$.*

Proof (of Claim 5). Suppose for contradiction that each vertex b in B dominates at most $2|\text{supp}(b)| + 3$ vertices of $T \cap N(u)$. Then, since every vertex a in $S \setminus B$ dominates at most 1 vertex of $T \cap N(u)$ by definition, we have

$$5k < |T \cap N(u)| \leq \sum_{b \in B} (2|\text{supp}(b)| + 3) + \sum_{a \in S \setminus B} 1 \leq 3k + 2 \sum_{b \in B} |\{a \mid a \prec b\}|$$
$$= 3k + 2 \sum_{a \in S} |\{b \mid a \prec b\}| \leq 3k + 2k = 5k$$

which is a contradiction. ◀

Let v be a vertex which dominates more than $2|\text{supp}(v)| + 3$ vertices of $T \cap N(u)$. Let us denote C the set of terminals that are common neighbors of u and v. We know, that there are $|C|$ eyes between u and v. For each internal eye which contains a vertex of S there is vertex in $\text{supp}(v)$ by Claim 4. Therefore, there are at most $|\text{supp}(v)| + 1$ eyes which contain a vertex of S inside. We show, that the eyes that do not contain any vertex of S do not contain any vertices at all.

Assume to the contrary that there is an eye u, x, v, y that does not contain any vertex of S, but its interior is not empty. If there was no terminal in the interior, then Reduction Rule 5 would apply, contradicting the instance being reduced with respect to the previous rules. Hence, the set of terminals inside the eye is non-empty. Let us denote it T_e. By Lemma 3, each terminal has to be dominated by a vertex of S. Since there are no vertices of S inside the eye and $u \notin S$ by assumption, it follows that $T_e \subseteq N(v)$ and Reduction Rule 4 would apply, again contradicting the instance being reduced with respect to the previous rules.

Hence, each eye not containing any vertex of S is empty. Therefore there are at most $|\text{supp}(v)| + 1$ nonempty eyes which have together at most $2|\text{supp}(v)| + 2$ terminals on their boundaries. Since C contains more than $2|\text{supp}(v)| + 3$ vertices it follows that there are two vertices x and y in C that only appear on boundaries of empty eyes and, hence, are of degree two. In particular, for their neighborhoods we have $N(x) = N(y) = \{u, v\}$ and Reduction Rule 3 would apply, contradicting the instance being reduced with respect to the previous rules. Hence, if the original instance was a yes-instance, then so is the resulting one, finishing the proof of the lemma. ◀

Equipped with Reduction Rule 6 it is easy to finally bound the number of terminals in the instance.

▶ **Reduction Rule 7.** If there is more than $5k^2$ terminals, then answer NO.

▶ **Lemma 10.** *Reduction Rule 7 is correct.*

Proof. Suppose for contradiction that $|T| > 5k^2$, but the instance is a yes instance. Let S be a solution. By Lemma 3 each vertex of T has at least one neighbor in S. On the other hand, $|S| \leq k$ and, hence, by pigeonhole principle there is a vertex in S with more than $5k$ neighbors in T. But this contradicts the instance being reduced with respect to Reduction Rule 6. ◀

2.3 Main Theorem and Time Complexity

In this subsection we piece together our main result. We start by analyzing the time needed to exhaustively apply the reduction rules.

▶ **Lemma 11.** *Given an instance (G, T, k) of* Planar Steiner Tree *one can in $O(n^4)$-time either correctly decide it or obtain an equivalent instance (G'', T', k') which is reduced with respect to Reduction Rules 1–7, $k' \leq k$, and $|T'| + k'$ is at most $|T| + k$.*

Proof. We apply the reduction rules exhaustively. The equivalence of resulting instance follows from the correctness of the reduction rules. Here we argue about the running time to apply the reduction rules.

Let us first consider the time spend to find one application and to apply the rule one time for each of the rules.

To check whether Reduction Rule 1 can be applied one can in linear time find the connected components of graph G and graph $G[T]$ and then check whether all the terminals are in the same connected components. Therefore, to check whether the rule applies we need $O(n)$ time. The rule can be applied in constant time, as we directly answer.

We can find an application of Reduction Rule 2 by going through the edges of the graph in linear time. The modification of the instance also takes linear time.

To check whether Reduction Rule 3 applies we can determine for every pair of vertices x, y whether their neighborhoods are equal in cubic time. The instance can again be modified in linear time.

To apply Reduction Rule 4 and Reduction Rule 5 we first need to find the embedding, the eyes, and which vertices are inside them. We try all pairs of non-terminals u, v and for each of them we find the eyes formed between them (if any) in linear time. Then in linear time we find for all other vertices inside which eye they are and finally whether the rules apply to any of the eyes. Summing over all pairs u, v, an application of the rule can be found in $O(n^3)$ time. The application takes constant time for Reduction Rule 4 and linear time for Reduction Rule 5.

An application of Reduction Rule 6 or Reduction Rule 7 can be found and the rule applied in linear time.

Now we would like to bound the number of times we search for an application of a rule and the number of times a rule is applied. Note that after any rule is applied as well as on the beginning we iterate through the reduction rules starting from the first one and continuing with the later ones and we stop when we find no application of a rule anymore. Hence the number of times we search for an application of a particular rule is linear in the number of applications of the rules.

If Reduction Rule 1 or 7 is applied, then the algorithms stops. Hence, these rules are only applied once. Reduction Rules 2, 3, and 5 reduce the number of vertices of the graph. Since no rule increases the number of vertices, there can be at most n application of these rules together. Finally, Reduction Rules 4 and 6 reduce the number of non-terminals (by converting them to terminals) and, hence, there can be at most n applications of these two rules together.

Summing up, the rules can be exhaustively applied in $O(n^4)$ time and the lemma follows by observing that no rule increases k or $k + |T|$. ◀

▶ Remark. It seems possible to improve the running time given in Lemma 11, e.g., Reduction Rule 3 only has to be applied for vertices of degree 2, etc. However, this would make analysis more complicated and we prefer to focus this extended abstract in different direction.

Now we are ready to prove our main theorem, which combines Lemma 11 with Proposition 1, to obtain the kernel for Planar Steiner Tree with respect to k.

▶ **Theorem 12.** *Given a Planar Steiner Tree instance (G, T, k) one can in $O(k^{284}n+n^4)$ time either correctly decide decide it or find an equivalent instance (G', T', k') with $k' \leq k$, $|T'| \leq k'^2$, and $|V(G')| \leq O(k^{284})$.*

Proof. By Lemma 11 we can in $O(n^4)$ time find an equivalent instance (G'', T', k') which is reduced with respect to Reduction Rules 1–7 and $k' \leq k$. Since the instance (G'', T', k') is reduced with respect to Reduction Rule 7, we have $|T'| \leq 5k^2$ and if (G'', T', k') is a yes-instance, then there is an optimal Steiner tree with at most $k' + |T'| = O(k'^2)$ edges.

Now we run the algorithm of Proposition 1 on the instance (G'', T') for $O(k'^{284}n)$ time. If it fails to finish, then the number of edges in an optimal Steiner tree k_{OPT} is more than $k' + |T'| = O(k'^2)$ and we answer NO. If it finishes, then let F be the set of edges returned. If $|F|$ is not $O(k'^{284})$, then we again answer NO. Otherwise, we let G' be the graph (W, F), where W is the set of vertices of G'' which are incident to at least one edge of F. By Proposition 1 and since no terminal is isolated in any Steiner tree, the instances (G'', T', k') and (G', T', k') are equivalent. Since $|F| = O(k_{\mathrm{OPT}}^{142})$ and $k_{\mathrm{OPT}} \leq 5k'^2 + k' = O(k'^2)$, the bound on the size of $V(G')$ follows. ◀

3 Conclusion and Future Directions

We presented the first polynomial kernel for Planar Steiner Tree with respect to the number k of Steiner points in the solution. It seems plausible that the kernel size bound as well as the running time of the algorithm can be improved. First of all, already the size bound $O((k+|T|)^{142})$ given by Proposition 1 probably offers plenty of space for improvements, with the best known lower bound (for the method) being just $\Omega((k+|T|)^2)$.

Second, our approach mostly relies on domination type arguments. Dominating Set in Planar Graphs is known to have a linear kernel [1] and even one such computable in linear time [3]. It might be possible to use the ideas of these kernels to reduce the number of terminals to linear in k or at least reduce the running time to linear. If one was able to reduce the number of terminals to linear in k, this would immediately lead to a subexponential FPT-algorithm for Planar Steiner Tree parameterized by k, which is itself an interesting open problem.

Given the amount of research conducted on Steiner Tree in general graphs with respect to the number of terminals $|T|$, it seems surprising that we cannot say anything more with respect to this parameterization on planar graphs. In particular, we are aware neither of polynomial kernel nor of a subexponential FPT-algorithm for Planar Steiner Tree with respect to this parameterization.

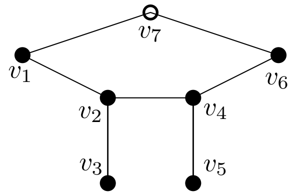

Figure 1 Illustration of different notions of optimality for Steiner forests. For $\mathcal{T} = \{\{v_1, v_6\}, \{v_2, v_3\}, \{v_4, v_5\}\}$, the set of terminals already induces a connected subgraph of G. Hence no Steiner point is needed. On the other hand, the spanning tree of this induced subgraph has 5 edges (and we cannot omit any of them), but it is possible to connect the terminals as required by taking the following 4 edges: $\{v_1, v_7\}, \{v_7, v_6\}, \{v_2, v_3\}, \{v_4, v_5\}$.

Finally, one might want to try to generalize the ideas of this paper to further problems and Planar Steiner Forest seems a natural candidate. Here we are given a graph G and a family of pairs of vertices $\mathcal{T}$ and the task is to find the smallest subgraph of G in which the vertices of each pair from $\mathcal{T}$ appear in the same connected component. While Pilipczuk et al. [36] proved an analogue of Proposition 1 for Planar Steiner Forest (with the degree of the polynomial in the bound 4 times larger), there are several issues preventing such a generalization.

First, for Steiner Forest it is no longer true that the optimal forest is the one obtained as a spanning forest of the subgraph with minimum number of Steiner points (usually all members of all pairs in $\mathcal{T}$ are called terminals), as Figure 1 illustrates. Since the result of Pilipczuk et al. minimizes the number of edges, whereas our formulation optimizes the number of Steiner points, it might be troublesome to combine them.

Second, for Steiner Forest there is no direct analogue of Reduction Rule 2. Indeed, notice that applying Reduction Rule 2 to, e.g., the edge $\{v_2, v_3\}$ of the graph on Figure 1 causes the optimum number of Steiner points in the solution to increase to one, as the vertex resulting from the contraction is no longer a terminal. Moreover the rule also applies to edge $\{v_1, v_2\}$ which is not in the Steiner forest minimizing the number of edges.

We still believe that one might be able to reduce the instance in such a way that the optimum solution only uses $O(k^2)$ edges in trees that contain at least two edges. It is not obvious whether such a reduction could be combined with the ideas of Pilipczuk et al. to obtain a polynomial kernel for Planar Steiner Forest with respect to the number of Steiner points in the solution.

There are also reasons not to believe that such a kernel should exist for Planar Steiner Forest, since the problem behaves significantly different than Planar Steiner Tree. Indeed, Pilipczuk et al. [36] also proved that, in contrast to Planar Steiner Tree, there is no subexponential FPT-algorithm for Planar Steiner Forest parameterized by the total number of edges in an optimal Steiner forest.

Hence, we believe that there are many interesting directions to study related to the current paper.

References

1 J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, May 2004.

2 M. W. Bern and P. E. Plassmann. The Steiner problem with edge lengths 1 and 2. *Inf. Process. Lett.*, 32(4):171–176, 1989.

3 R. van Bevern, S. Hartung, F. Kammer, R. Niedermeier, and M. Weller. Linear-time computation of a linear problem kernel for dominating set on planar graphs. In *Parameterized and Exact Computation - 6th International Symposium, IPEC 2011, Revised Selected Papers*, volume 7112 of *LNCS*, pages 194–206. Springer, 2011.
4 A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the 39th ACM Symposium on Theory of Computing, STOC 2007*, pages 67–74. ACM, 2007.
5 H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015.
6 H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
7 G. Borradaile, E. D. Demaine, and S. Tazari. Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs. *Algorithmica*, 68(2):287–311, 2014.
8 G. Borradaile, P. N. Klein, and C. Mathieu. An $O(n \log n)$ approximation scheme for Steiner tree in planar graphs. *ACM Transactions on Algorithms*, 5(3), 2009.
9 J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6:1–6:33, Feb. 2013.
10 D. Cieslik. *Steiner minimal trees*, volume 23. Springer Science & Business Media, 1998.
11 M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012*, pages 74–84. IEEE, 2012.
12 M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pages 150–159. IEEE Computer Society, 2011.
13 M. Dom, D. Lokshtanov, and S. Saurabh. Kernelization lower bounds through colors and IDs. *ACM Transactions on Algorithms*, 11(2):13:1–13:20, 2014.
14 R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.
15 S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1:195–207, 1972.
16 R. E. Erickson, C. L. Monma, and A. F. Veinott, Jr. Send-and-split method for minimum-concave-cost network flows. *Mathematics of Operations Research*, 12(4):634–664, 1987.
17 F. V. Fomin, F. Grandoni, D. Kratsch, D. Lokshtanov, and S. Saurabh. Computing optimal Steiner trees in polynomial space. *Algorithmica*, 65(3):584–604, 2013.
18 F. V. Fomin, P. Kaski, D. Lokshtanov, F. Panolan, and S. Saurabh. Parameterized single-exponential time polynomial space algorithm for Steiner tree. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Proceedings, Part I*, volume 9134 of *LNCS*, pages 494–505. Springer, 2015.
19 F. V. Fomin, D. Lokshtanov, and S. Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 142–151. SIAM, 2014.
20 D. S. R. Frank K. Hwang and P. Winter. *The Steiner Tree Problem*, volume 53 of *Annals of Discrete Mathematics*. Elsevier, 1992.
21 B. Fuchs, W. Kern, D. Mölle, S. Richter, P. Rossmanith, and X. Wang. Dynamic programming for minimum Steiner trees. *Theory of Computing Systems*, 41(3):493–500, 2007.
22 M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem in NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.

23 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

24 S. L. Hakimi. Steiner's problem in graphs and its implications. *Networks*, 1:113–133, 1971.

25 M. Hauptmann and M. Karpinski. A compendium on Steiner tree problems. online. `http://theory.cs.uni-bonn.de/info5/steinerkompendium/`.

26 J. Hopcroft and R. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, Oct. 1974.

27 F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem.* North-Holland, Amsterdam, 1992.

28 M. Jones, D. Lokshtanov, M. S. Ramanujan, S. Saurabh, and O. Suchý. Parameterized complexity of directed Steiner tree on sparse graphs. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Proceedings*, volume 8125 of *LNCS*, pages 671–682. Springer, 2013.

29 A. B. Kahng and G. Robins. *On Optimal Interconnections for VLSI.* Kluwer Academic Publisher, 1995.

30 P. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms*, 19(1):104 – 115, 1995.

31 B. Korte, H. J. Prömel, and A. Steger. Steiner trees in VLSI-layout. In *Paths, Flows and VLSI-Layout*, pages 185–214, 1990.

32 A. Y. Levin. Algorithm for the shortest connection of a group of graph vertices. *Sov. Math. Dokl.*, 12:1477–1481, 1971.

33 J. Nederlof. Fast polynomial-space algorithms using inclusion-exclusion. *Algorithmica*, 65(4):868–884, 2013.

34 M. Patrascu and R. Williams. On the possibility of faster SAT algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010*, pages 1065–1075. SIAM, 2010.

35 M. Pilipczuk, M. Pilipczuk, P. Sankowski, and E. J. van Leeuwen. Subexponential-time parameterized algorithm for Steiner tree on planar graphs. In *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013*, volume 20 of *LIPIcs*, pages 353–364. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

36 M. Pilipczuk, M. Pilipczuk, P. Sankowski, and E. J. van Leeuwen. Network sparsification for Steiner problems on planar and bounded-genus graphs. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 276–285. IEEE Computer Society, 2014.

37 H. J. Prömel and A. Steger. *The Steiner Tree Problem; a Tour through Graphs, Algorithms, and Complexity.* Vieweg, 2002.