Nearly ETH-tight algorithms for Planar Steiner Tree with Terminals on Few Faces*

Sándor Kisfaludi-Bak† Jesper Nederlof‡ Erik Jan van Leeuwen§

Abstract

The Steiner Tree problem is one of the most fundamental NP-complete problems as it models many network design problems. Recall that an instance of this problem consists of a graph with edge weights, and a subset of vertices (often called terminals); the goal is to find a subtree of the graph of minimum total weight that connects all terminals. A seminal paper by Erickson et al. [Math. Oper. Res., 1987] considers instances where the underlying graph is planar and all terminals can be covered by the boundary of k faces. Erickson et al. show that the problem can be solved by an algorithm using $n^{O(k)}$ time and $n^{O(k)}$ space, where n denotes the number of vertices of the input graph. In the past 30 years there has been no significant improvement of this algorithm, despite several efforts.

In this work, we give an algorithm for Planar Steiner Tree with running time $2^{O(k)}n^{O(\sqrt{k})}$ using only polynomial space. Furthermore, we show the running time of our algorithm is almost tight: we prove that there is no $f(k)n^{o(\sqrt{k})}$ algorithm for Planar Steiner Tree for any computable function f, unless the Exponential Time Hypothesis fails.

1 Introduction

In the Steiner Tree problem, we are given an undirected n-vertex graph G with edge weights[1] $\omega : E(G) \to \{0, \dots, W\}$ and a set of *terminals* $T \subseteq V(G)$. We are asked to find an edge set S (called a *Steiner tree*) minimizing $\sum_{e\in S} \omega(e)$ such that every two vertices $u, v \in T$ are connected in the graph (V, S). The problem is one of the most important NP-complete problems as it elegantly models network design problems. Several textbooks are entirely devoted to Steiner trees [18, 40].

Parameterization by Number of Terminals A very popular research direction that aims to understand the computational complexity of Steiner Tree is to consider its parameterization by the number of terminals of the instance. Dreyfus and Wagner [17] and independently by Levin [29] initiated this line of research and showed that the problem can be solved in $3^{|T|}\texttt{poly}(n)$ time.[2] Thus, in the language of *parameterized complexity*, Dreyfus and Wagner show the problem is Fixed Parameter Tractable when parameterized by $|T|$. Fuchs et al. [25] improved this result to $O^*(c^{|T|})$ for any $c > 2$. In the case of small weights, Björklund et al. [6] provide a faster $O^*(2^{|T|}W)$ time algorithm. All aforementioned algorithms require almost as much working memory as time. However, the setting in which one is given only working memory that is polynomial in the input size has also been well-studied [20, 21, 30, 36]. The currently fastest polynomial-space algorithms run in $O^*(2^{|T|}W)$ time [30] and $O^*(7.97^{|T|})$ time [21].

Planar Steiner Tree Another very popular direction is to study Steiner Tree restricted to *planar graphs*. The study of approximation schemes for Planar Steiner Tree (and many variations and generalizations of it) has been a well-established subject for a long time [7, 8, 2, 1]. More recently, our understanding of the exact exponential-time complexity of Planar Steiner Tree has also progressed significantly. Some positive results study the decision variant of the unweighted case of Planar Steiner Tree, and its parameterization by $|S|$ (the size of the required Steiner tree). Pilipczuk et al. [39] show that one can preprocess the input instance in polynomial time to remove all but $O(|S|^{142})$ edges. Pilipczuk et al. [38] (and later, Fomin et al. [22]) show the problem can be solved in $O^*(2^{\sqrt{|S|\log^2|S|}})$ time. The square-root in the exponent is typical for exact algorithms for problems on planar graphs (intuitively, due to the planar separator theorem), and is often called the 'square-root phenomenon'. However, such a running time is not always guaranteed. Very recently [34], it was shown that when parameterized by the number of terminals $|T|$, planarity probably gives little advantage over the algorithm of Dreyfus and Wagner in the following strong sense: if Planar Steiner Tree can be solved in $O^*(2^{o(|T|)})$, then the Exponential Time Hypothesis fails.

Planar Steiner Tree with Terminals on Few Faces A broadly studied variant of Planar Steiner

*Supported by the Netherlands Organization for Scientific Research NWO under project no. 024.002.003 and 639.021.438.

†Eindhoven University of Technology, The Netherlands; s.kisfaludi.bak@tue.nl

‡Eindhoven University of Technology, The Netherlands; j.nederlof@tue.nl

§Utrecht University, The Netherlands; e.j.vanleeuwen@uu.nl

[1] For convenience, we assume all comparisons and additions of weights take $O(1)$ time.

[2] In this paper we use the $O^*(\cdot)$ which omits factors polynomial in the input size.

TREE is obtained by making assumptions on the locations of the terminals. Such natural assumptions are also studied extensively in e.g. the classic flow paper by Ford and Fulkerson [23]. Of particular interest is the case when all terminals lie on k given faces of the plane-embedded input graph G. This parameter has a long history in the study of cuts and (multicommodity) flows (e.g. [35, 10, 28, 27]) and shortest paths (e.g. [24, 11]). Krauthgamer et al. [27] (in this SODA) dubbed it the *terminal face cover number* $\gamma(G)$. The case $\gamma(G) = 1$ is known as an Okamura-Seymour graph [37]. For PLANAR STEINER TREE, the parameterization by $k = \gamma(G)$ generalizes the parameterization by $|T|$, as we can always ensure that $k \leq |T|$. Hence, this parameterization generalizes both previous research directions.

An important result by Erickson et al. [19] shows that the problem can be solved in $n^{O(k)}$ time. Their algorithm for $k = 1$ arises in both the aforementioned approximation algorithms (i.e. in spanner constructions [7]) and fixed-parameter algorithms (i.e. in preprocessing algorithms [39]). Hence, the algorithm plays a central role in the literature on PLANAR STEINER TREE.

The quest to improve, refine, and generalize the result by Erickson et al. [19] received significant attention. Bern [3, 31] improved the constant in the exponent of the running time of [19] to 2, and gave a better running time if many terminals do not share any face with other terminals. Bern and Bienstock [4] studied a generalization in which the terminals can be removed by removing k consecutive outerplanar layers. Provan [41, 42] studied generalizations in which covering by faces is replaced with covering by 'path-convex regions', motivated by some problems in geometry. For an excellent survey of previous work on PLANAR STEINER TREE with terminals on a few faces we refer to [26, Chapter 5], or to [12].

Despite these previous studies going back over 30 years, all previously known algorithms use $n^{\Omega(k)}$ time, and the algorithms matching this time bound use $n^{\Omega(k)}$ space. From a lower bound perspective, the result by Marx et al. [34] implies that no $O^*(2^{o(k)})$-time algorithm exists assuming the Exponential Time Hypothesis (as we can always ensure that $k \leq |T|$). However, this still leaves a large gap between the lower and the upper bound. This leads to the natural question what the true computational complexity is of PLANAR STEINER TREE with terminals on k faces.

1.1 Our Results In this work we almost settle the exact complexity of PLANAR STEINER TREE parameterized by the number of faces needed to cover the terminals, modulo the Exponential Time Hypothesis. First, we show that the algorithm of Erickson et al. [19] can be significantly improved:

THEOREM 1.1. *Given a plane n-vertex graph G with terminals T, edge weights $\omega : E(G) \to \{0, \ldots, W\}$ and a set $\mathcal{K} \subseteq 2^{E(G)}$ of k faces of G such that each vertex from T is on a face in $\mathcal{K}$, a minimum weight Steiner tree can be found using $2^{O(k)} n^{O(\sqrt{k})} \log W$ time, and polynomial space.*

Observe that our algorithm uses only polynomial space, in contrast to all previous algorithms with a running time of the type $n^{O(k)}$.

We remark that we may assume that the planar embedding and faces are not a priori given, as already observed in previous work. Explicitly motivated by our setting, Bienstock and Monma [5] showed that, given only the graph, one can find k faces covering all terminals in some embedding in $2^{O(k)}\texttt{poly}(n)$ time. Hence, we can simply run their algorithm on the input graph before applying Theorem 1.1 without affecting the bound on the running time.

We also remark that Marx et al. [34] recently gave an algorithm for PLANAR STEINER TREE with running time $n^{O(\sqrt{|T|})}W$. Note that $k \leq |T|$ and $|T|$ can be arbitrary large when $k = 1$, but nevertheless our result is incomparable to theirs because of the $2^{O(k)}$ factor in our running time.

We complement our algorithm with a conditional lower bound that almost (that is, modulo the $2^{O(k)}$ factor) matches the running time of our algorithm:

THEOREM 1.2. *There is no $f(k)n^{o(\sqrt{k})}$ algorithm for* PLANAR STEINER TREE *for any computable function f, unless the Exponential Time Hypothesis fails.*

In terms of parameterized complexity, this theorem implies that PLANAR STEINER TREE is $W[1]$-hard when parameterized by the number of terminal faces.

1.2 Our Techniques We describe our techniques along with intuition and relationship to previous works.

Our Algorithm Before we sketch our algorithm, we sketch the previous work we build upon. A simple observation behind the known exact algorithms for STEINER TREE (all the way back to [17, 29]) is that any edge e of the solution S splits S into two subtrees S_1 and S_2; if we know e and which terminal is connected in which subtree, we can simply recursively solve the associated subproblems (or look up their solutions in a Dynamic Programming table). The number of candidates for e and the split of the terminal set is $|E| \cdot 2^{|T|}$, which is (roughly) the running time of [17, 29] and their refinements.

The algorithm by [19] builds upon this scheme and additionally uses the following observation. Suppose

$T' = \{t_1, \ldots, t_p\} \subseteq T$ are terminals that lie on a single face numbered in cyclic order along the face, and $i < j < k < l$ such that t_i and t_k are connected in S_1 and t_j and t_l are connected in S_2, then S_1 and S_2 intersect and thus we can remove an edge and obtain another solution S' with $\omega(S') \leq \omega(S)$. Hence, we can restrict our attention to subproblems in which terminal sets form an interval on each face. The number of candidates for e and such terminal sets is $|E| \cdot n^{2k}$, which is (roughly) the running time of [19] and their refinements.

Our approach fits into the general scheme of guessing how separators based on a solution map into an input (see also, for example, [34, 33]). For the aimed running time $2^{O(k)} n^{O(\sqrt{k})}$, we cannot afford the above decomposition as S_1 and S_2 may interact in k faces from $\mathcal{K}$. Instead, we use a larger separator on S to decompose S in two forests S_1 and S_2 such that only a few faces from $\mathcal{K}$ intersect both S_1 and S_2. To this end, our *crucial idea* is to consider a separator in the graph $H = S \cup \mathcal{K}^\flat$, where $\mathcal{K}^\flat$ denotes $\cup_{K \in \mathcal{K}} K$. We show that H has a (balanced) separator X of size $O(\sqrt{k})$, and if we consider the split of S into S_1 and S_2 that X induces on S, we see that any face in $\mathcal{K}$ not intersecting X is either entirely connected in S_1 or in S_2. Algorithmically, this observation allows us to guess the set X, and a partition of the faces from $\mathcal{K}$ to be covered in both subproblems which we solve recursively. Faces from $\mathcal{K}$ intersecting with X can still be connected both via S_1 and S_2, so their terminals set still needs to be distributed, but by the observation of [19] we can restrict attention to slits induced by intervals.

Our Lower Bound Our lower bound builds on ideas of the recent $2^{\Omega(|T|)}\texttt{poly}(n)$ lower bound by Marx et al. [34], but instead of reducing from 3-SAT, we reduce from the Grid Tiling problem. An instance of Grid Tiling consists of two integers n and k, and k^2 sets or *cells* $M_{a,b} \subseteq [n] \times [n]$ for $a, b \in [k]$, and we are tasked to decide whether there exist integers $x_a \in [n]$ and $y_a \in [n]$ for $a \in [k]$ such that $(x_a, y_b) \in M_{a,b}$ for all $a, b \in [k]$. The standard way to do a reduction from Grid Tiling in geometric problems (see e.g. [32]) is to have a gadget for each cell that is capable of representing the choice (x_a, y_b) in that cell, and designing some communication gadget, which when applied to horizontally (or vertically) neighboring cells, ensures that the first (resp. second) index of the choices in these cells are equal. One of the main challenges is to design these communication gadgets, capable of transmitting $\log n$ bits of information between any pair of neighboring cells in the grid of $k \times k$ cells.

Our main innovation in the lower bound is the design of a novel communication gadget, the so-called *flower gadget*, that is capable of communicating multiple bits, but uses only a single terminal face. Essentially, we need a gadget with $2n$ portal vertices with the property that in any optimal solution, the Steiner tree will have exactly two components induced by the gadget, rooted at portal vertex i and $n + i$ respectively for some $i \in \{1, \ldots, n\}$. This already prescribes a rotational symmetry to the gadget, but it is challenging to find a gadget that can support this for several reasons. In particular, a pair of canonical trees within the gadget must contain an equal number of terminals, and together they must contain all terminals of the terminal face. The easiest way to ensure that the root of one tree uniquely determines the root of the other is to ensure that the root of the tree uniquely determines an interval of terminals from the face that need to be contained in the tree. In practice, this is enforced by making sure that the root of the tree has degree two, and a canonical tree has a (subdivided) binary tree structure. Such a branching structure can be enforced in a continuous setting if one places the roots in the hyperbolic plane on a circle of radius r, and the terminals on a concentric circle of radius $2r$. This was the inspiration for the construction. In some weak sense, our gadget models such a metric space with a rolled up Euclidean grid and a special weighting scheme. Note that the construction itself does not use any tools from hyperbolic geometry, it is completely elementary. An important feature of the weighting is that it can be de-weighted: by replacing every edge by a path whose length is the weight of that edge, we obtain an unweighted graph of polynomial size.

Organization In Section 2 we define some necessary notation and folklore results we will use. Section 3 is devoted to the proof of Theorem 1.1, while Section 4 is devoted to the proof of Theorem 1.2. In Section 5 we briefly summarize our paper and point out opportunities for further research.

2 Preliminaries

For a set X, we let $\Pi(X)$ denote the set of all partitions of X. If $\pi \in \Pi(X)$, we write π is a partition *of* X. We call a partition π *finer* than partition π', and denote this relation by $\pi \preceq \pi'$, if for every $u, v \in X$, u and v are in the same block in π implies that u and v are in the same block in π'. Given two partitions π, π' we use the notation $\pi \sqcup \pi'$ for the join in the partition lattice, that is, the finest partition that coarsens both π and π'. If $\pi \in \Pi(X)$ and $\pi \in \Pi(Y)$, we also define the join $\pi \sqcup \pi' \in \Pi(X \cup Y)$ obtained by adding singletons to π and π' to make them elements of $\Pi(X \cup Y)$. If $u, v \in X$, we write $\{\{u, v\}\}$ for the partition in which all elements except u and v are in singleton blocks. If $W \subset X$, and $\pi \in \Pi(X)$ we let $\pi_{|W}$ be the projection of π on W, that is two elements are in the same block of $\pi_{|W}$ if and only if there are in the same block in π.

If $G = (V, E)$ is a graph, we write $V(G) := V$ and $E(G) := E$. If $S \subseteq E(G)$, we denote $V(S) := \bigcup_{\{u,v\}\in S}\{u, v\}$ for the set of vertices incident to edges of S. For a vertex subset $S \subset V$, let $N[S]$ be the closed neighborhood of S, that is, the set S together with all vertices adjacent to a vertex in S. If $X \subseteq E$, we denote $N_X[v]$ to be all vertices sharing an edge in X with v. A vertex subset $X \subseteq V(G)$ is called a *dominating set* if $N[X] = V(G)$. If G is connected, a vertex $v \in V(G)$ is called an *articulation vertex* if $G[V \setminus \{v\}]$ is not connected. A graph is 2-*connected* if it does not contain any articulation vertex. We say that a path P in G is a *maximal* 2-*path* if all internal vertices of P have degree 2 and its ends have degree not equal to 2. Note that all maximal 2-paths in G are edge disjoint.

If $\mathcal{F} \subseteq 2^U$, we use the notation $\mathcal{F}^\flat := \bigcup_{F\in\mathcal{F}} F$ for its flattening.

Treewidth and Balanced Separators We will use the fact that the treewidth of a plane graph and its dual graph are closely related, which follows from the planar grid minor theorem. In particular, we use the following sharp result:

THEOREM 2.1. ([9]) *For every plane graph G with dual graph G^*, $|\mathrm{tw}(G) - \mathrm{tw}(G^*)| \leq 1$.*

The following well-known theorem follows in a standard fashion from the grid minor theorem (our particular statement follows from combining Theorem 7.23 from [13] with Lemma 3.1 in [14]).

THEOREM 2.2. ([15]) *If a graph G is planar and has a dominating set of size k, then $\mathrm{tw}(G) \leq 15\sqrt{k}$.*

DEFINITION 2.1. (BALANCED SEPARATION) *A pair of vertex subsets (Y, Z) is a* separator *in graph G if $Y \cup Z = V(G)$ and there are no edges in G between $Y \setminus Z$ and $Z \setminus Y$. For a fixed weight function $w : V(G) \to \mathbb{R}$, we say that a separation (Y, Z) is a w-*weighted α-balanced separation *in G if $w(Y \setminus Z) \leq \alpha \cdot w(V(G))$ and $w(Z \setminus Y) \leq \alpha \cdot w(V(G))$.*

LEMMA 2.1. (LEMMA 7.20 FROM [13]) *Suppose G has treewidth* tw*, and consider a nonnegative function $w : V(G) \to \mathbb{R}_{\geq 0}$. Then G has a $\frac{2}{3}$-balanced separation (A, B) of order at most* $\mathrm{tw} + 1$.

3 Algorithm for PLANAR STEINER TREE with terminals on few faces

This section is devoted to the proof of Theorem 1.1. Refer to Section 1.2 for a high level description and intuition. To simplify the analysis of our algorithm, we show (in the full version) that w.l.o.g. the degree of each vertex is at most 3 and the graph is 2-connected.

LEMMA 3.1. *Let G be a plane graph with terminals T, edge weights $\omega : E(G) \to \{0, \ldots, W\}$, and a set $\mathcal{K} \subseteq 2^{E(G)}$ of k faces of G such that each vertex from T is on a face in $\mathcal{K}$. Then one can compute in polynomial time a 2-connected subcubic planar graph G' with terminals T', edge weights $\omega' : E(G') \to \{0, \ldots W\}$, and a set $\mathcal{K}' \subseteq 2^{E(G')}$ of k faces of G' such that each vertex in T' is on a face in $\mathcal{K}'$ and moreover, any Steiner tree in G corresponds to a Steiner tree in G' of the same weight and any Steiner tree S' in G' corresponds to a Steiner tree in G of weight at most $\omega'(S')$.*

We are now ready to describe the algorithm. Since we present a recursive algorithm, it is more convenient to work with a slightly more general problem that is solved in recursive steps. We first define this more general problem.

DEFINITION 3.1. (BLOCK STEINER FOREST) *Given nonempty subsets $B, T \subseteq V(G)$, and a partition π of B, we say $S \subseteq E(G)$ is a (G, B, π, T)-*Block Steiner Forest *if in $(V(G), S)$*

(a) every vertex in T is connected to at least one vertex in B, and

(b) a pair of vertices in B are connected if and only if they are in the same block of π.

PLANAR BLOCK STEINER FOREST (PBSF)
Instance: A plane graph G, weights $\omega : E(G) \to \{0, \ldots, W\}$, subsets $B, T \subseteq V(G)$, partition π of B.
Asked: The minimum $\omega(S)$ where S is a (G, B, π, T)-Block Steiner Forest S.

In the above problem we can think of the vertices in B as *boundary vertices*. Typically, other parts of the Steiner tree will intersect only in B and already establish some connectivity, which allows us to only connect vertices in B according to π.

We need to establish the following for the base case of the algorithm.

LEMMA 3.2. *For any constant c_0 there is an algorithm* `steinerBase`$(G, \omega, B, \pi, T, \mathcal{K})$ *that solves a Block Steiner Forest instance (G, B, π, T) in polynomial time, provided that $|B| + |\mathcal{K}| \leq c_0$.*

Before we prove Lemma 3.2, we need to introduce the following tools.

THEOREM 3.1. ([19]) *Let (G, ω, T) be a given instance of* PLANAR STEINER TREE*, and let $\mathcal{K}$ be a given set of $O(1)$ faces such that $T \subseteq V(\mathcal{K})$. Then the instance can be solved in polynomial time.*

Definition 3.2. (Non-crossing sequence) *A sequence $x \in [\ell]^n$ is* non-crossing *if for all $1 \leq i < j < k < l \leq n$, we have that $x_i \neq x_k$, $x_j \neq x_l$, or x_i, x_j, x_k, x_l are all equal. It is* minimal *if there is no i such that $x_i = x_{i+1} = x_{i+2}$.*

Lemma 3.3. *The length of a minimal non-crossing sequence with ℓ different values is at most 4ℓ.*

Proof. Use induction on ℓ. For $\ell = 1$ the statement trivially holds, so assume $\ell > 1$. For a value $v \in [\ell]$, denote $l(v)$ (and $r(v)$) the smallest i (and largest) i such that $x_i = v$. As any two intervals $[l(v), r(v)]$ and $[l(v'), r(v')]$ are either disjoint or contained in each other, we can always find an interval that does not contain any other interval. Then necessarily $r(v) = l(v)+1$ or $r(v) = l(v)$. Consider the sequence obtained by removing indices $l(v), r(v)$. If this leads to a quadruple or triple of consecutive equal values, then remove at most two of them so that exactly two remain. We obtain a minimal non-crossing sequence on $\ell - 1$ values which has length at most $4(\ell-1)$ by induction. Since we have removed at most four indices, the lemma follows. □

Proof of Lemma 3.2. Let $S \subseteq E(G)$ be an optimal solution. Then there exists a partition π_S of B such that $\pi \preceq \pi_S$ and two vertices u, v of B are in the same connected component of $G[S]$ if and only if $\{\{u, v\}\} \preceq \pi_S$. By enumerating all possibilities, we may by abuse of notation assume that $\pi = \pi_S$. This adds a constant factor to the running time, as $|B| \leq c_0$.

Let $\{B_1, \ldots, B_\ell\}$ be the partition π, and let $S_1, \ldots, S_\ell$ be the corresponding subtrees of S. Let $K \in \mathcal{K}$ have terminals $t_1, \ldots, t_p$, enumerated in order of appearance of a walk on the face (since G is 2-connected, the face boundary forms a cycle). Suppose that $i < j < k < l$, and terminals t_i and t_k are connected in S and terminals t_j and t_l are connected in S. Then all four terminals must be connected to each other in a tree S_z as they all lie on the same face. Thus, if for each terminal in t_i we let $x_i \in [\ell]$ encode the index of the block it is connected to within S, then x is a non-crossing sequence.

Such a non-crossing sequence can be encoded by its minimal non-crossing sequence (obtained by removing all but two elements from each subsequence of the same element) and a mapping from the indices of the non-crossing sequence to $V(K) \cap T$. As the length of the minimal non-crossing sequence is at most 4ℓ by Lemma 3.3, there are at most $\ell^{4\ell} n^{4\ell}$ different sequences x.

The algorithm now is as follows: enumerate all possible combinations of sequences $x_K \in [\ell]^{|V(K)\cap T|}$ for each face $K \in \mathcal{K}$. Then for each $i \in [\ell]$ solve the instance (G, ω, T_i) using the algorithm of Theorem 3.1, where $T_i = \bigcup_{K \in \mathcal{K}} x_K^{-1}(i)$. The running time is polynomial since $\ell \leq |B|$ and $|\mathcal{K}|$ are constants. □

Our main effort in the remainder of this section will be to prove the following lemma, of which Theorem 1.1 is an easy consequence (we postpone the proof of Theorem 1.1 to the end of this section). By Lemma 3.1, we may assume the input graph is 2-connected and subcubic.

Lemma 3.4. *Suppose (G, ω, B, T, π) is an instance of Planar Block Steiner Forest, and $\mathcal{K}$ is a set of faces of G such that $T \subseteq V(\mathcal{K})$. Then Algorithm* `steiner`$(G, \omega, B, \pi, T, \mathcal{K})$ *as listed in Algorithm 1 correctly solves the* PBSF *instance (G, B, π, T).*

We continue with the description of the procedure `steiner` as listed in Algorithm 1. For a subset $X \subseteq V(G)$, we let $\mathcal{K}(X) \subseteq \mathcal{K}$ be the set of faces from $\mathcal{K}$ whose edges intersect with X. For a face $K \in \mathcal{K}$ and a vertex set $X \subseteq V(G)$, let $\mathrm{cc}(K, X) \in \Pi(V(K) \setminus X)$ be the partition of the face vertices induced by removing X, that is, $\mathrm{cc}(K, X)$ is the collection of vertex sets of the connected components of the subgraph of $(V(K), E(K))$ induced by $V(K) \setminus X$.

In Line 1 we use Lemma 3.2 as the base case. At Line 3 we guess what the separator X is (as already described in Section 1.2), and at Line 4 we guess how the boundary vertices and faces not intersecting X are distributed among the subproblems. At Line 7 we guess for each segment of faces from $\mathcal{K}$ obtained after removing X whether they are connected in the first or second subproblem. Note that these segments do not include X itself. Based on all these guesses we compute the set of terminals T_1 and T_2 to be connected in both subproblems on Line 8 and Line 9. At Line 10 we guess what connectivity is established in both problems (encoded as partitions), and on Line 12 we check whether the two partitions jointly encode all required connectivity.

Proof of Lemma 3.4. We need to establish that the algorithm gives a feasible solution that is optimal, which we do in two steps.

Correctness: Feasibility Let $\mathsf{Best} :=$ `steiner`$(G, \omega, B, \pi, T, \mathcal{K})$. We prove that $\omega(S) \leq \mathsf{Best}$ for some (G, B, π, T)-Block Steiner Forest S. The base case follows from Lemma 3.2. For the recursive case, consider the iteration at Line 15 where Best is updated for the last time. By induction there is a $(G, B_1 \cup X, \pi_1, T_1)$-Block Steiner Forest S_1 such that $\omega(S_1) \leq \mathsf{Best}_1$, and a $(G, B_2 \cup X, \pi_2, T_2)$-Block Steiner Forest S_2 such that $\omega(S_2) \leq \mathsf{Best}_2$.

We claim that $S := S_1 \cup S_2$ is a (G, B, π, T)-Block Steiner Forest. Note that two vertices in $B_1 \cup B_2 \cup X$

Algorithm $\texttt{steiner}(G, \omega, B, \pi, T, \mathcal{K})$

Output: Minimum $\omega(S)$ over all (G, B, π, T)-Block Steiner Forests $S \subseteq E(G)$.

1: **If** $|B| + |\mathcal{K}(T)| \le c_0$ **then return** $\texttt{steinerBase}(G, \omega, B, \pi, T, \mathcal{K})$

2: $\mathsf{Best} \leftarrow \infty$

3: **for** every $X \in \binom{V(G)}{\le 15\sqrt{|\mathcal{K}(T)| + |B|} + 2}$ **do**

4: **for** $B_1 \subseteq B \setminus X$, and $\mathcal{K}_1 \subseteq \mathcal{K}(T) \setminus \mathcal{K}(X)$ such that $\frac{|B_1| + |\mathcal{K}_1|}{|B| + |\mathcal{K}(T)|} \in [\frac{1}{3}, \frac{2}{3}]$ **do**

Based on the above guessed split of boundary vertices and terminal faces for the first subproblem, compute the corresponding sets for the second subproblem

5: $B_2 \leftarrow (B \setminus X) \setminus B_1$

6: $\mathcal{K}_2 \leftarrow (\mathcal{K}(T) \setminus \mathcal{K}(X)) \setminus \mathcal{K}_1$

Try all subsets of segments of terminal faces to assign to the first subproblem

7: **for** all $\mathcal{A}_1 \subseteq \bigcup_{K \in \mathcal{K}} \mathrm{cc}(K, X)$ **do**

Compute the terminal sets for both subproblems based on the above guesses

8: $T_1 \leftarrow (\mathcal{A}_1^\flat \cap T) \cup \bigcup_{K \in \mathcal{K}_1} (V(K) \cap T)$

9: $T_2 \leftarrow (V(\mathcal{K}(X)^\flat) \setminus X) \setminus \mathcal{A}_1^\flat \cup \bigcup_{K \in \mathcal{K}_2} (V(K) \cap T)$

10: **for** all partitions π_1 on $B_1 \cup X$ and partitions π_2 on $B_2 \cup X$ **do**

11: $\pi' \leftarrow \pi_1 \sqcup \pi_2$

Check if the two partitions implement the required connectivity

12: **if** $\pi'_{|B} = \pi$ **and** for all $u \in X$, there exists $v \in B$ with $\{\{u, v\}\} \preceq \pi'$ **then**

Solve the subproblems recursively, and update current minimum if needed

13: $\mathsf{Best}_1 \leftarrow \texttt{steiner}(G, \omega, B_1 \cup X, \pi_1, T_1 \setminus X, \mathcal{K}_1 \cup \mathcal{K}(X))$

14: $\mathsf{Best}_2 \leftarrow \texttt{steiner}(G, \omega, B_2 \cup X, \pi_2, T_2 \setminus X, \mathcal{K}_2 \cup \mathcal{K}(X))$

15: $\mathsf{Best} \leftarrow \min\{\mathsf{Best}, \mathsf{Best}_1 + \mathsf{Best}_2\}$

16: **return** Best

Algorithm 1: Algorithm implementing Theorem 1.1.

are connected in S if and only if they are in the same block of $\pi' = \pi_1 \sqcup \pi_2$. Therefore, as we require that $(\pi_1 \sqcup \pi_2)_{|B} = \pi$ on Line 12, S satisfies property (b) of Definition 3.1. To see that S also satisfies (a) of Definition 3.1, consider some terminal $t \in T$. We distinguish three cases:

- If $t = u \in X$, then u is connected to a vertex $v \in B$ as we require $\{\{u, v\}\} \preceq \pi_1 \sqcup \pi_2$ for some $v \in B$ on Line 12.
- If $t \notin X$, and $\mathcal{K}(t) \in \mathcal{K}(X)$, then t will be either in T_i for $i = 1$ or $i = 2$, depending on whether the member of $\mathrm{cc}(K, X)$ containing t is in $\mathcal{A}_1$ or not, and by induction t will be connected to some vertex $u \in B_i \cup X$ in (V, S_i). If t is connected to a vertex in B_i, then we are done as $B_i \subseteq B$; if t is connected to a vertex in X, then the first case applies.
- If $t \notin X$, and $\mathcal{K}(t) \in \mathcal{K}_i$ for $i \in \{1, 2\}$ then either at Line 8 or Line 9 we add t to T_i, and by induction t will be connected to some vertex in $B_i \cup X$ in (V, S_i). If t is connected to a vertex in B_i, then we are done as $B_i \subseteq B$; if t is connected to a vertex in X, then the first case applies.

Thus S is a (G, B, π, T)-Block Steiner Forest. The claim follows as

$$\omega(S) \le \omega(S_1) + \omega(S_2) \le \mathsf{Best}_1 + \mathsf{Best}_2 \le \mathsf{Best}.$$

Correctness: Optimality Denote $\mathsf{Best} := \texttt{steiner}(G, \omega, B, \pi, T, \mathcal{K})$. We prove that $\omega(S) \ge \mathsf{Best}$ for every (G, B, π, T)-Block Steiner Forest S. We do this by showing that there exists some partition S_1, S_2 of S such that in some iteration $\omega(S_1) \ge \mathsf{Best}_1$ and $\omega(S_2) \ge \mathsf{Best}_2$. Note that since $\omega(e) \ge 0$ for every $e \in E(G)$, we may assume that S is a forest: if S would have a cycle, then we could remove any edge on the cycle to obtain a new S with less or equal weight.

Consider the subgraph $H = (V(S) \cup V(\mathcal{K}(T)) \cup B, S \cup \mathcal{K}(T)^\flat)$ of G, with the embedding inherited from the embedding of G. Let H^* be the planar dual of H; that is, for every face of H we create a vertex H^*, and two vertices in H^* are connected with an edge in H^* if and only if the corresponding faces in H share an edge.

CLAIM 3.1. $\mathrm{tw}(H) \le 15\sqrt{|\mathcal{K}(T)|} + 1$.

Proof. If $\mathcal{K}(T) = \emptyset$, then H is a forest and $\mathrm{tw}(H) = 1$. Otherwise, we claim that $\mathcal{K}(T) \subseteq V(H^*)$ is a dominating set of H^*. To see this, note that removing the edges of a face in the primal H amounts to contracting the neighborhood of the corresponding vertex in the dual to a single vertex. Therefore we know that if we contract the sets $N_{H^*}[v]$ for all $v \in \mathcal{K}(T)$, we are left with a single vertex (being the planar dual of a forest). This implies that $\mathcal{K}(T)$ is a dominating set of H^*: if there is a vertex in $V(H^*) \setminus N_{H^*}[\mathcal{K}]$, it would still be a vertex in the graph after contracting and there would be at least two vertices in the planar dual of the forest, which is a contradiction.

Now we use Lemma 2.2 to obtain $\mathrm{tw}(H^*) = 15\sqrt{|\mathcal{K}(T)|}$. As H^* is the dual of H, by Theorem 2.1 we have the bound $\mathrm{tw}(H) \le 15\sqrt{|\mathcal{K}(T)|} + 1$. ☐

Now we consider the following weight function $w : V(H) \to \mathbb{N}$:

- For every $v \in V(G)$, set $w(v) = 0$
- For every face K in $\mathcal{K}(T)$, arbitrarily pick a vertex $v \in V(K)$ (which could be in B) and set $w(v) = 1$,
- For every $v \in B$, set $w(v) = w(v) + 1$.

By Claim 3.1 and Lemma 2.1 there is a w-weighted $\frac{2}{3}$-balanced separation (Y, Z) in H such that $|Y \cap Z| \le 15\sqrt{k} + 2$. In some iteration of the loop at Line 3, the algorithm will set $X = Y \cap Z$. Since the separation (Y, Z) of H is balanced with respect to w, we have that $w(Y), w(Z) \le \frac{2}{3}w(V(G))$. This implies that $\frac{|B \cap (Y \setminus X)| + |\mathcal{K}(Y) \setminus \mathcal{K}(X)|}{|\mathcal{K}(T)| + |B|} \in [\frac{1}{3}, \frac{2}{3}]$. Therefore, the algorithm will set $B_1 = (B \cap Y) \setminus X$ and $\mathcal{K}_1 = \mathcal{K}(Y) \setminus \mathcal{K}(X)$ in some iteration of the loop at Line 4.

Note that in this iteration we set $B_2 = (B \setminus X) \setminus B_1$ which equals $(B \cap Z) \setminus X$, since $B \subseteq Y \cup Z$. Similarly, we set $\mathcal{K}_2 = ((\mathcal{K}(T) \setminus \mathcal{K}(X)) \setminus \mathcal{K}_1$.

Note that $\mathcal{K}_2 = \mathcal{K}(Z) \setminus \mathcal{K}(X)$, because if a face has vertices from both Y and Z, then it must also have vertices from X as (X, Y) is a separation and every face is a cycle (this follows as G is 2-connected, see i.e. [16, Theorem 4.2.5]). Moreover, if two vertices $a, b \notin X$ are in the same connected component $C \in \mathrm{cc}(K, X)$, then either both are in Y or both are in Z, as (Y, Z) is a separation of a graph containing the edge set K.

Thus, at some iteration of the loop at Line 7 the algorithm will set $\mathcal{A}_1$ such that for every face $K \in \mathcal{K}(X)$, any connected component C of $\mathrm{cc}(K, X)$ is contained in Y if it is in $\mathcal{A}_1^\flat \cap V(K)$, and it is contained in Z otherwise. It follows that there is some iteration in which $T_1 \setminus X = (Y \setminus X) \cap T$ and $T_2 \setminus X = (Z \setminus X) \cap T$.

The separation (Y, Z) of H induces a partition of S into two subforests S_1, S_2, where $S_1 = \{\{u, v\} \in S : \{u, v\} \in Y\}$ and $S_2 := S \setminus S_1$ (note that we add edges of S contained in $E(X)$ to S_1 and not to S_2).

Let π_1 be the partition on $B_1 \cup X$ where $\{u, v\} \in B_1 \cup X$ are in the same block of π_1 if and only if u and v are connected in the graph (V, S_1). Similarly, let π_2 be the partition on $B_2 \cup X$ where $\{u, v\} \in B_2 \cup X$ are in the same block of π_2 if and only if u and v are connected in the graph (V, S_2).

Since $S = S_1 \cup S_2$ is a (G, B, π, T)-Block Steiner Forest, we see that $\pi_1 \sqcup \pi_2$ equals π, and that it satisfies the conditions checked on Line 12. As the algorithm loops over all options of π_1, π_2 on Line 10, eventually it will try the pair π_1, π_2.

We can conclude that S_1 is a $(G, B_1 \cup X, \pi_1, T_1)$-Block Steiner Tree, and that S_2 is a $(G, B_2 \cup X, \pi_2, T_2)$-Block Steiner Tree. As $\mathcal{K}_1 \subseteq \mathcal{K}(T)$ and $\mathcal{K}_2 \subseteq \mathcal{K}(T)$ we have by induction that $\omega(S_1) \ge \mathsf{Best}_1$ and, $\omega(S_2) \ge \mathsf{Best}_2$. Now $\omega(S) = \omega(S_1) + \omega(S_2) \ge \mathsf{Best}_1 + \mathsf{Best}_2 \ge \mathsf{Best}$ follows. This concludes the proof. ☐

Proof of Theorem 1.1. Arbitrarily pick a terminal $t_0 \in T$. Then $\mathtt{steiner}(G, \omega, \{t_0\}, \{\{t_0\}\}, T_0 \setminus t_0, \mathcal{K})$ will be the minimum weight of a tree connecting T by Lemma 3.4, which is exactly what needs to be computed in the PLANAR STEINER TREE instance. Since G is subcubic, every vertex is in at most three faces of $\mathcal{K}$ and thus, $|\mathcal{K}(X)| + |X| \le 4|X|$. Therefore, if $|\mathcal{K}| + |B|$ is larger than some constant c_0, then in a recursive call with parameters $\mathcal{K}', B'$ we have

$$\begin{aligned} |\mathcal{K}'| + |B'| &\le \tfrac{2}{3}(|\mathcal{K}| + |B|) + 4(15\sqrt{|\mathcal{K}| + |B|} + 2) \\ &\le \tfrac{3}{4}(|\mathcal{K}| + |B|). \end{aligned}$$

Thus the recursion depth of $\mathtt{steiner}$ is at most $O(\log |\mathcal{K}|)$, and $|B| = O(\sqrt{k} \log k)$ for any recursive call.

If we let $T(n, p)$ denote the running time of $\mathtt{steiner}$ when $|B| + |\mathcal{K}| = p$ we see that

$$T(n, p) = \begin{cases} n^{O(1)} & \text{if } p \text{ is constant} \\ n^{O(\sqrt{p})} 2^{O(p)} T(n, \frac{3}{4}p) & \text{otherwise.} \end{cases}$$

To see this, note the loop at Line 3 has at most $n^{O(|X|)}$ iterations; the loop at Line 4 has at most $2^{|B| + |\mathcal{K}|}$ iterations, the loop on Line 7 has at most $2^{|X|}$ iterations (as $|\mathrm{cc}(K, X)| \le |X|$), and the loop on Line 10 has at most $(|B| + |X|)^{O(|B| + |X|)}$ iterations (as

$|\pi(B)| = |B|^{O(|B|)}$). All other operations (apart from the recursion) require polynomial time. This recurrence solves to $2^{O(p)} n^{O(\sqrt{p})}$, thus the theorem follows. $\square$

4 Lower Bound

In this section, we aim to prove Theorem 1.2. Throughout, for any integer n, let $[n] = \{1, \ldots, n\}$ (where $[0] = \emptyset$) and let $[n] \times [n] = \{(1,1), \ldots, (1,n), (2,1), \ldots, (n,n)\}$.

We present a reduction from Grid Tiling, which is defined as follows. An instance of Grid Tiling consists of two integers n and k, and k^2 sets $M_{a,b} \subseteq [n] \times [n]$ for $a, b \in [k]$. Let $\mathcal{M} = \{M_{a,b} \mid a, b \in [k]\}$. Since n and k can be derived by inspecting $\mathcal{M}$, we may specify the instance by $\mathcal{M}$ alone. The Grid Tiling problem asks to decide whether there exist integers $x_a \in [n]$ and $y_a \in [n]$ for $a \in [k]$ such that $(x_a, y_b) \in M_{a,b}$ for all $a, b \in [k]$. In this case, we call $x_1, \ldots, x_k, y_1, \ldots, y_k$ a solution to the instance. The following statement is known for Grid Tiling.

Theorem 4.1. ([32, 13]) *There is no $f(k) \cdot n^{o(k)}$-time algorithm for* Grid Tiling *for any computable function f, unless the Exponential Time Hypothesis fails.*

The reduction follows from the following theorem.

Theorem 4.2. *Let $\mathcal{M}$ be an instance of* Grid Tiling, *with associated integers n and k. Then in time polynomial in n and k, one can construct an integer $K_{\mathcal{M}}$ a planar graph $\mathcal{G}_{\mathcal{M}}$ and set $T_{\mathcal{M}}$ of terminals such that*

- *$\mathcal{G}_{\mathcal{M}}$ has size $O(k^{16} n^{27})$;*
- *$T_{\mathcal{M}}$ can be covered by $k(k-1)+1$ faces of an embedding of $\mathcal{G}_{\mathcal{M}}$;*
- *$\mathcal{M}$ admits a solution if and only if $\mathcal{G}_{\mathcal{M}}$ admits a Steiner tree of at most $K_{\mathcal{M}}$ edges.*

The construction draws on ideas from Marx et al. [34], but differs in crucial points. A sequence of the elaborate verification gadgets by Marx et al. [34] is capable of communicating while relying on only constantly many terminals as long as the Steiner tree connects these gadgets in a chain. In essence, the gadgetry can be used to represent the choices in a single row of the $k \times k$ grid, and ensure the communication. In addition, each copy of the verification gadget is capable of extracting 1 bit of information vertically. Unfortunately such gadgetry cannot be used for communicating both horizontally and vertically in the $k \times k$ grid, since the chaining property would mean that the Steiner tree would have to induce cycles. To get around this problem, Marx et al. designs a "connector gadget" that can transmit one bit of information vertically, without introducing connectivity. The gadget uses four terminals adjacent that are on one face. Since we would need a large number of terminal faces to communicate the required bits vertically, the connector gadget does not yield a parameterized reduction for our parameter k. In order to extract multiple bits, we modify the gadget sequence into n-tuples of verification gadgets; this only leaves open the issue of communicating $\omega(1)$ bits of information without connectivity, and using only $O(1)$ terminal faces.

Given Theorem 4.2, Theorem 1.2 is quickly proven.

Proof of Theorem 1.2. Suppose there is an $f(k)n^{o(\sqrt{k})}$-time algorithm $\mathcal{A}$ for Planar Steiner Tree for some computable function f. We now construct a fast algorithm for Grid Tiling. Let $\mathcal{M}$ be an instance of Grid Tiling, with associated integers n and k'. Apply Theorem 4.2 to $\mathcal{M}$, which takes time polynomial in n and k', and let $K_{\mathcal{M}}$ be the resulting integer, $\mathcal{G}_{\mathcal{M}}$ the resulting planar graph, and $T_{\mathcal{M}}$ the corresponding terminal set. Run $\mathcal{A}$ on $\mathcal{G}_{\mathcal{M}}$ and let S denote the resulting Steiner tree. Answer "yes" if S has at most $K_{\mathcal{M}}$ edges, and answer "no" otherwise. This completes the description of the algorithm.

The correctness of the algorithm is immediate from the third item of Theorem 4.2. Since $T_{\mathcal{M}}$ can be covered by $k'(k'-1)+1$ faces of an embedding of $\mathcal{G}_{\mathcal{M}}$ and $\mathcal{G}_{\mathcal{M}}$ has size $O((k')^{16} n^{27})$, the algorithm runs in $f'(k')n^{o(k')}$ time for some computable function f'. According to Theorem 4.1, this implies that the Exponential Time Hypothesis fails. $\square$

We now set out to prove Theorem 4.2. Throughout, let N, L, t be large integers to be chosen later. Let M be a large integer such that $M > 10 \cdot NL$ and let $M_i = M^i$. The construction consists of two types of gadgets. The first is the flower gadget (the main novelty of our construction), the second are the verification gadgets. We now present both types in detail and discuss their properties. Then we show how these gadgets can be brought together and prove Theorem 4.2.

4.1 Flower Gadget

The gadget is based on a "rolled" grid whose edges have a special weighting scheme. It is easier to study the unrolled version of the grid first, which we do in Section 4.1.1. We establish some properties of the metric space induced by the weighting, and prove the key statement (Lemma 4.1) that we need for Steiner trees in this unrolled context. Then, in Section 4.1.2, we present the full rolled gadget and prove the essential properties we need in the final construction.

4.1.1 The unrolled grid

For $a, b \in \mathbb{Z}$, the discrete interval with endpoints a, b is the set $\{a, a+1, \ldots, b\}$, which we denote by $[\![a,b]\!]$. The discrete intervals form a poset with respect to the containment relation: let Γ be the (undirected) Hasse diagram of this poset. Equivalently, Γ is the subgraph of the square grid restricted to integer points on or above the line $y-x=0$. In this diagram, we can talk about ancestors and descendants; in particular, for any pair of points p and q the lowest common ancestor (the smallest interval that contains both p and q) is well-defined.

We define a weight function w on the edges of Γ in the following way. For an edge pq, let $q=[\![a,b]\!]$ be the larger discrete interval, that is, suppose $p \subset q$. Then the weight $w(pq)$ is set to $2^{-\lfloor \log(b-a) \rfloor}$. See Figure 1.

We introduce some more terminology and notation. For discrete intervals that are singletons, we use the shorthand $[\![a]\!]=[\![a,a]\!]$. A *monotone path* in Γ is a path $p_1, p_2, \dots, p_k$ where $p_1 \subset p_2 \subset \cdots \subset p_k$. For a pair of intervals that share an endpoint, the monotone path between them is unique. The *triangle* of a discrete interval $q=[\![a,b]\!]$ is the set of its subintervals; we denote this set by $\Delta(q)$ or $\Delta([\![a,b]\!])$. The lowest common ancestor of the intervals p and q is denoted by $p \wedge q$. Let dist denote the shortest path distance in Γ, i.e., $\operatorname{dist}(a,b) := \inf\{\sum_{pq \in P} w(pq) \mid P \text{ is a path from a to b}\}$. For a vertex subset $S \subset V(\Gamma)$, let $\operatorname{dist}(p,S)=\inf_{s \in S} \operatorname{dist}(p,s)$; the distance of subsets of $V(\Gamma)$ is their Hausdorff distance. A *layer* is a maximal subset of edges in Γ of the same weight, i.e., layer i has weight 2^{-i}. See Figure 1 for an illustration of the weights.

For any $x \in \mathbb{Z}$, the *vertical* at $x/2$ is $V_{x/2}=\{[\![a,b]\!] \mid \frac{a+b}{2}=\frac{x}{2}\}$. Note that if $p=[\![a,b]\!]$, then $p \in V_{(a+b)/2}$; also, if x is a multiple of 2, then $[\![x/2]\!] \in V_{x/2}$. A *column* is the set of edges between two consecutive verticals $V_{x/2}$ and $V_{(x+1)/2}$. For any $y \in \mathbb{N}$, the *horizontal* at y is $H_y=\{[\![a,b]\!] \mid b-a=y\}$. Note that $H_0=\{[\![a]\!] \mid a \in \mathbb{Z}\}$. A *row* is the set of edges between two consecutive horizontals H_y and H_{y+1}, and the *height* of an edge is the index of the horizontal passing through its lower endpoint. Notice that the weight of edges is weakly decreasing as the height is increasing.

Proposition 4.1. *If $p \subseteq q$ are discrete intervals, then any monotone path between them is a shortest path. If $p=[\![a,b]\!]$ is a discrete interval and $x \in \mathbb{Z}$, then the distance from p to the vertical $V_{x/2}$ is realized by the straight monotone path from p to its lowest ancestor in $V_{x/2}$. Finally, if p and q are incomparable, then the union of the straight paths $p \to (p \wedge q)$ and $(p \wedge q) \to q$ is a shortest path from p to q.*

Proof. For the first claim, if $p \subset q$, then any path between them must contain edges that traverse from a discrete interval of size $|p|$ to a discrete interval of size $|p|+1$, an edge from size $|p|+1$ to $|p|+2$, etc., and an edge from size $|q|-1$ to $|q|$. Any monotone path contains only one of each edge listed. Furthermore, edges where the intervals have identical size have identical weight. Hence, all monotone paths are shortest paths.

For the second claim, suppose without loss of generality that $x \geq a+b$. Let $v \in V_{x/2}$ be a vertex where $\operatorname{dist}(p, V_{x/2}) = \operatorname{dist}(p,v)$ (see Figure 2). Since the distance from a point to a set is defined as an infimum, we first show that such a vertex minimizing the distance exists. Note that all but finitely many vertices of $V_{x/2}$ are ancestors of p, and among the ancestors the closest one is $[\![a, x-a]\!]$ since it minimizes the set of rows that it needs to pass. Therefore, the minimum distance is either realized by $[\![a,x-a]\!]$ or a non-ancestor of p, of which there are finitely many; consequently, such a vertex v exists, and if it is an ancestor of p, then it must be $[\![a,x-a]\!]$.

Suppose that the straight path from p to $[\![a,x-a]\!]$ is not a shortest path. We now consider the case where v is incomparable to $[\![a,b]\!]$. A path from p to a vertex v in $V_{x/2}$ that is incomparable to p must traverse an edge in each vertical column between $V_{(a+b)/2}$ (the vertical containing p) and $V_{x/2}$. Among the paths that contain exactly one edge of each of these columns and no other edges, we know that the straight path from p to $[\![a,x-a]\!]$ is shortest, since all of its edges have maximal height, and therefore minimal possible weight; this is a contradiction.

If $x/2 > b$, then V is disjoint from $\Delta(p)$, that is, it does not contain any descendants of p, therefore the only remaining possible distance minimizing vertex is $[\![a,x-a]\!]$. Otherwise, if v is a descendant of p, then by the first claim of this lemma, any monotone path is a shortest path, and in particular, the vertex $[\![x-b,b]\!]$ minimizes the distance to p among the descendants of p in V (by minimizing the set of rows that it needs to cross). But the unique monotone path from p to $[\![x-b,b]\!]$ is at least as long as $\operatorname{dist}(p,[\![a,x-a]\!])$, since both shortest paths have exactly one edge from each column between $V_{(a+b)/2}$ and $V_{x/2}$, but in each column the edge in the path from p to $[\![a,x-a]\!]$ is higher and therefore has less or equal weight; this is a contradiction. Therefore, $v=[\![a,x-a]\!]$ is a closest neighbor of p in V, and the shortest path is a straight monotone path due to the first claim of this lemma.

For the third claim, if p and q are incomparable, then let $[\![a,b]\!]=p$ and $[\![a',b']\!]=q$. Without loss of generality, $a' > b$. Let V be the vertical passing through $p \wedge q$, that is, let $V=V_{(a+b')/2}$. Note that $[\![a,b]\!] \wedge [\![a',b']\!] = [\![a,b']\!] \in V$. Since V separates p and q, we have that $\operatorname{dist}(p,q) \geq \operatorname{dist}(p,V) + \operatorname{dist}(V,q)$. By the previous claim, we have $\operatorname{dist}(p,V)=\operatorname{dist}(p,[\![a,((a+b')-a)]\!])=\operatorname{dist}(p,p\wedge q)$. Then, by symmetry, we have

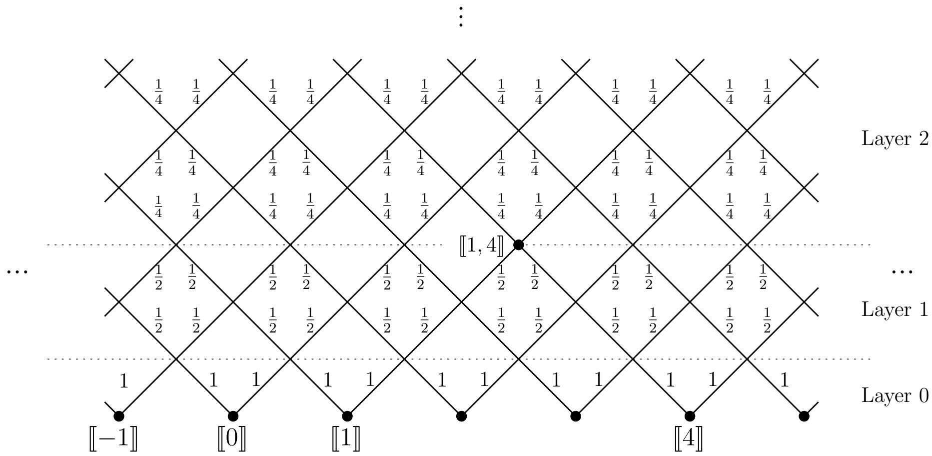

Figure 1: The Hasse diagram Γ of the poset of discrete intervals. Indicated are the weights assigned by the weight function w as well as the different layers.

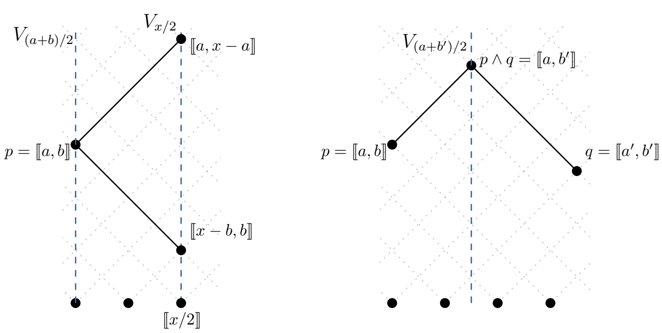

Figure 2: Left: Distance from p to a vertical. Right: The distance between a pair of non-comparable points p and q.

$\mathrm{dist}(q, S) = \mathrm{dist}(q, p \wedge q)$, and the claim holds. □

The *left diagonal* at b is $LD_b = \{⟦x, b⟧ \mid x \in \mathbb{Z}\}$, the *right diagonal* at a is $RD_a = \{⟦a, x⟧ \mid x \in \mathbb{Z}\}$. Figure 3 shows examples of both.

Proposition 4.2. *The distance of LD_b and $V_{b+1/2}$ is 1. The distance of LD_a and RD_{a+1} is 2.*

Proof. There is a shortest path from $⟦a, b⟧$ to $V_{b+1/2}$ that is monotone and has $b - a + 1$ edges and ends at $⟦a, 2b + 1 - a⟧$ by Proposition 4.1. We claim that all of these paths have weight exactly 1. We use induction on $b - a$. The path from $⟦b⟧$ to $⟦b, b+1⟧$ has a single edge of weight 1. Consider the shortest path starting at $⟦a, b⟧$. By induction the shortest path starting at $⟦a + 1, b⟧$ has weight 1, so it is sufficient to show that they have equal weight. Notice that these paths traverse the same horizontal edge rows, except the row from size $|b - a - 1|$ to $|b - a|$ that is only traversed by the path of $⟦a + 1, b⟧$, and the rows from size $|2b - 1 - 2a|$ to $|2b + 1 - 2a|$ that are only traversed by the path of $⟦a, b⟧$. Notice that the edge row unique to the path of $⟦a + 1, b⟧$ is in layer $\lfloor \log(b - a) \rfloor$, while the two edges unique to the path of $⟦a, b⟧$ are both in layer $\lfloor \log(2b - 2a) \rfloor = \lfloor \log(2b - 2a + 1) \rfloor$. Consequently, the edge unique to $⟦a + 1, b⟧$ is precisely one layer below the two edges unique to $⟦a, b⟧$, and thus the two paths have equal weight.

To prove the claim about the distance of diagonals, we can apply the first claim: $\mathrm{dist}(LD_a, RD_{a+1}) \leq \mathrm{dist}(LD_a, V_{a+1/2}) + \mathrm{dist}(V_{a+1/2}, RD_{a+1})$, where both terms on the right hand side are 1 by the first claim. On the other hand, there is a path of length 2: the path $⟦a⟧; ⟦a, a + 1⟧; ⟦a + 1⟧$. □

Lemma 4.1. *Let p be a discrete interval and let $\ell > 0$. The weight of any Steiner tree for the terminal set $\{p, ⟦0⟧, \ldots, ⟦\ell⟧\}$ is at least $2\ell + \mathrm{dist}(\Delta(⟦0, \ell⟧), p)$.*

Proof. Let $p = ⟦a, b⟧$ and suppose without loss of generality that $a + b \geq \ell$, that is, p is on or to the right of $V_{\ell/2}$. The proof is by double induction, first on ℓ, and second for a fixed ℓ on the distance $\mathrm{dist}(\Delta(⟦0, \ell⟧), p)$. Clearly for $\ell = 0$, the Steiner tree is at least as long as the distance from $⟦0⟧$ to p. Let $\ell \geq 1$, and let S be the Steiner tree. We distinguish several cases based on the location of p, see Figure 3.

Case 1. $p \in \Delta(⟦0, \ell⟧)$, (that is, $a \geq 0$ and $b \leq \ell$) If p has degree 1, then let q be the nearest vertex within S to p that has degree at least 3. The tree S is also a Steiner tree for the terminal set $\{q, ⟦0⟧, \ldots, ⟦\ell⟧\}$, and $\mathrm{dist}(\Delta(⟦0, \ell⟧), p) = 0 \leq \mathrm{dist}(\Delta(⟦0, \ell⟧), q)$, so it is sufficient to prove the claim for q instead of p. Therefore, without loss of generality, assume that p has degree at least 2. Let r be a neighbor of p. The edge pr defines two subtrees rooted at p: one where the shortest path to p traverses pr and one where it does not. Each tree must contain some non-empty subinterval of the terminals $[\![0]\!], \dots, [\![\ell]\!]$; suppose that S_1 contains $[\![0]\!], \dots, [\![x]\!]$ and S_2 contains $[\![x+1]\!], \dots, [\![\ell]\!]$.

By induction on ℓ, we have that

$$\begin{aligned} w(S) &\geq w(S_1) + w(S_2) \\ &\geq 2x + \operatorname{dist}(p, \Delta([\![0,x]\!])) + 2(\ell - x - 1) + \\ &\qquad \operatorname{dist}(p, \Delta([\![x+1,\ell]\!])) \\ &\geq 2\ell - 2 + \operatorname{dist}(\Delta([\![0,x]\!]), \Delta([\![x+1,\ell]\!])) \\ &\geq 2\ell - 2 + \operatorname{dist}(LD_x, RD_{x+1}) \\ &\geq 2\ell, \end{aligned}$$

where the last inequality follows from Proposition 4.2.

Case 2. $p \notin \Delta([\![0,\ell]\!])$, (that is, $a < 0$ or $b < \ell$) Without loss of generality, assume that p is a vertex of $V(S)$ that maximizes $\operatorname{dist}(p, \Delta([\![0,\ell]\!]))$. (Note that for a given tree S, the lemma gives the strongest lower bound for such a vertex p). Furthermore, among vertices maximizing this distance, there must be at least one vertex p with a neighbor q where $\operatorname{dist}(q, \Delta([\![0,\ell]\!])) < \operatorname{dist}(p, \Delta([\![0,\ell]\!]))$. Suppose there is no such vertex p; then let p be an arbitrary distance-maximizing vertex. There is a path from p to $[\![0]\!]$ where p is at positive distance from $\Delta([\![0,\ell]\!])$, while $[\![0]\!]$ is at distance 0. So there is an edge $p'q'$ on the path where $\operatorname{dist}(q', \Delta([\![0,\ell]\!])) < \operatorname{dist}(p', \Delta([\![0,\ell]\!])) = \operatorname{dist}(p, \Delta([\![0,\ell]\!]))$; this is a contradiction.

So we can suppose without loss of generality that p has a neighbor q such that $\operatorname{dist}(q, \Delta([\![0,\ell]\!])) < \operatorname{dist}(p, \Delta([\![0,\ell]\!]))$. If p has degree 1, then by induction on the distance we have that

$$\begin{aligned} w(S) &= w(pq) + w(S \setminus \{pq\}) \\ &\geq w(pq) + 2\ell + \operatorname{dist}(q, \Delta([\![0,\ell]\!])) \\ &\geq 2\ell + \operatorname{dist}(p, \Delta([\![0,\ell]\!])). \end{aligned}$$

Suppose now that p has degree at least 2. Similarly to Case 1, we define the trees S_1 containing $[\![0]\!], \dots, [\![x]\!]$ and S_2 containing $[\![x+1]\!], \dots, [\![\ell]\!]$ based on the branching at p. By induction, we have

$$\begin{aligned} w(S) &\geq w(S_1) + w(S_2) \\ &\geq 2x + \operatorname{dist}(p, \Delta([\![0,x]\!])) + 2(\ell - x - 1) \\ &\qquad + \operatorname{dist}(p, \Delta([\![x+1,\ell]\!])) \\ &= 2\ell - 2 + \operatorname{dist}(p, \Delta([\![0,x]\!])) \\ &\qquad + \operatorname{dist}(p, \Delta([\![x+1,\ell]\!])) \end{aligned} \tag{4.1}$$

It remains to show that $\operatorname{dist}(p, \Delta([\![0,x]\!])) + \operatorname{dist}(p, \Delta([\![x+1,\ell]\!])) \geq 2 + \operatorname{dist}(p, \Delta([\![0,\ell]\!]))$.

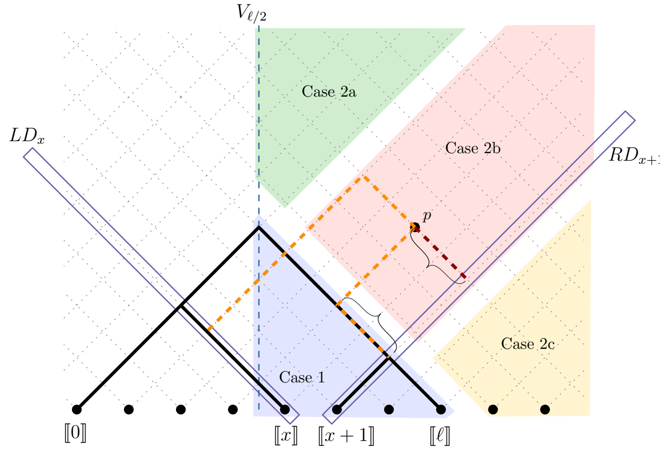

Figure 3: The case distinction of Lemma 4.1 with Case 2b illustrated.

Case 2a. $[\![0,\ell]\!] \in \Delta(p)$, (that is, $a \leq 0$ and $b > \ell$) We take shortest paths from p to $\Delta([\![0,x]\!])$ and $\Delta([\![x+1,\ell]\!])$ as suggested by Proposition 4.1. Note that $[\![0,\ell]\!] \in \Delta(p)$ implies that all vertices in both triangles (and also in $\Delta([\![0,\ell]\!])$ are descendants of p, so the distance can be realized from p to any of the three triangles is realized by an arbitrary monotone path from p to the tip of the triangle (this path traverses the least amount of horizontal rows). In particular, we can use an arbitrary monotone path P from p to $[\![0,\ell]\!]$ together with the monotone path P_x from $[\![0,\ell]\!]$ to $[\![0,x]\!]$ to realize $\operatorname{dist}(p, \Delta([\![x+1,\ell]\!]))$, and we can use P with the monotone path P_{x+1} from $[\![0,\ell]\!]$ to $[\![x+1,\ell]\!]$ to realize $\operatorname{dist}(p, \Delta([\![x+1,\ell]\!]))$. Therefore,

$$\begin{aligned} &\operatorname{dist}(p, \Delta([\![0,x]\!])) + \operatorname{dist}(p, \Delta([\![x+1,\ell]\!])) \\ &\quad = \quad 2w(P) + w(P_x) + w(P_x + 1) \\ &\quad = \quad 2\operatorname{dist}(p, \Delta([\![0,\ell]\!])) + \operatorname{dist}([\![0,x]\!], [\![0,\ell]\!]) \\ &\qquad\qquad + \operatorname{dist}([\![0,\ell]\!], [\![x+1,\ell]\!]) \\ &\quad \geq \quad 2\operatorname{dist}(p, \Delta([\![0,\ell]\!])) + \operatorname{dist}(LD_x, RD_{x+1}) \\ &\quad \geq \quad \operatorname{dist}(p, \Delta([\![0,\ell]\!])) + 2, \end{aligned} \tag{4.2}$$

where the last inequality uses Proposition 4.2. Then (4.1) and (4.3) combined imply that $w(S) \geq 2\ell + \operatorname{dist}(p, \Delta([\![0,\ell]\!]))$, as claimed.

Case 2b. p and $[\![0,\ell]\!]$ are incomparable, and $[\![x+1,\ell]\!] \in \Delta(p)$ (that is, $0 < a \leq x+1$) As previously, it is sufficient to show that $\operatorname{dist}(p, \Delta([\![0,x]\!])) + \operatorname{dist}(p, \Delta([\![x+1,\ell]\!])) \geq 2 + \operatorname{dist}(p, \Delta([\![0,\ell]\!]))$. Let P_1 be the unique monotone path from p to $[\![a,\ell]\!]$ and let P_2 be the unique monotone path from $[\![a,\ell]\!]$ to $[\![x+1,\ell]\!]$. By Proposition 4.1, $P_1 \cup P_2$ realizes the distance from p to $[\![x+1,\ell]\!]$. Let P_x be a shortest path from p to $\Delta([\![0,x]\!])$. Next,

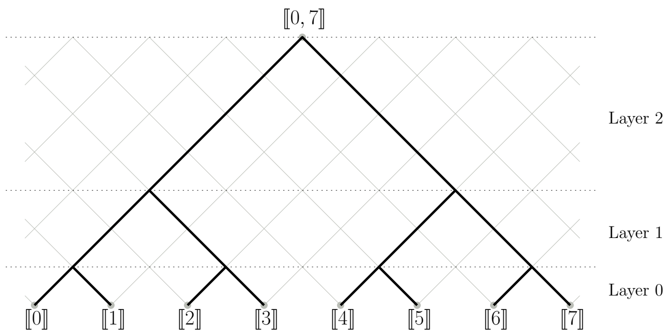

Figure 4: Optimal Steiner tree for the terminal set $\{[\![0, 2^3-1]\!], [\![0]\!], \ldots, [\![2^3-1]\!]\}$.

we replace P_2 by P_3 which is defined as the unique monotone path from p to $[\![x+1, b]\!]$. Notice that P_2 and P_3 has the same number of edges, but the edges in P_3 are higher and therefore $w(P_3) \leq w(P_2)$. So we have that $\mathrm{dist}(p, \Delta([\![0, x]\!])) + \mathrm{dist}(p, \Delta([\![x+1, \ell]\!])) = w(P_x) + w(P_1) + w(P_2) \geq w(P_x) + w(P_3) + w(P_1)$. Note that $P_x \cup P_3$ is a path from LD_x to RD_{x+1}, so it has length at least 2, and P_1 is a path from p to $\Delta([\![0, \ell]\!])$, so its length is at least $\mathrm{dist}(p, \Delta([\![x+1, \ell]\!]))$.

Case 2c. p and $[\![x+1, \ell]\!]$ are incomparable (that is, $a > x+1$)
We again need that $\mathrm{dist}(p, \Delta([\![0, x]\!])) + \mathrm{dist}(p, \Delta([\![x+1, \ell]\!])) \geq 2 + \mathrm{dist}(p, \Delta([\![x+1, \ell]\!]))$. The intervals in the area between LD_x and RD_{x+1} are all between p and $\Delta([\![0, x]\!])$, so any shortest path from p to $\Delta([\![0, x]\!])$ contains a subpath from LD_x to RD_{x+1}, and therefore has length at least 2. The shortest path from p to $\Delta([\![x+1, \ell]\!])$ is also a path from p to $\Delta([\![0, \ell]\!])$ since $\Delta([\![x+1, \ell]\!]) \subset \Delta([\![0, \ell]\!])$; therefore, its length is at least $\mathrm{dist}(p, \Delta([\![0, \ell]\!]))$. □

Let $\ell = 2^k - 1$ for some positive integer k. Observe that there is a tree for the terminal set $\{[\![0, \ell]\!], [\![0]\!], \ldots, [\![\ell]\!]\}$ of weight exactly 2ℓ (which is shortest possible by Lemma 4.1). It is easy to show by induction that there is a tree mimicking a binary tree that contains $2^k - i$ monotone paths traversing layer i for each $i = 0, \ldots, k-1$; all such paths have weight 1. See Figure 4 for an example.

4.1.2 Construction and properties of the flower gadget

The flower gadget is a finite planar graph sharing many properties of Γ. We give two equivalent definitions. Let $t \geq 4$ be a power of 2.

The first definition is to restrict Γ to the set $\{[\![a, b]\!] \mid b - a \leq t/2 - 1,\ a \geq 0,\ b \leq t\}$, and identify the vertex pairs $(0, b)$ and (t, b') where $b' = b + t$ for all $b = 1, \ldots, t$. Let Γ_t be the resulting weighted graph. The alternative and more intuitive definition requires the introduction of discrete intervals modulo t. Let $a, b \in \{0, 1, \ldots, t-1\}$. The discrete interval $[\![a, b]\!]_t$ is defined as $\{a, a+1, \ldots, b\}$ if $a \leq b$ or as $\{a, \ldots, t-1, 0, \ldots, b\}$ otherwise. Then Γ_t is the Hasse diagram for the poset of discrete intervals modulo t of size at most $t/2$, i.e., the Hasse diagram of the set

$$\Big\{[\![a, b]\!]_t \,\Big|\, a, b \in \{0, 1, \ldots, t-1\}, \\ (a \leq b \wedge b - a \leq t/2 - 1) \\ \vee (b < a \wedge t + b - a \leq t/2 - 1)\Big\}.$$

The weighting is identical to Γ: the weight of an edge pq where $p \subset q = [\![a, b]\!]_t$ is $2^{-\lfloor \log((b-a) \bmod t) \rfloor}$. Note that Γ_t is planar, since it can clearly be drawn on a cylinder, which is topologically equivalent to a punctured plane. See Figure 5 for a planar embedding.

The terminals of the flower gadget are its discrete intervals of size 1, and the portal vertices are the maximal discrete intervals in the poset. The portals will be used to connect a flower gadget to the rest of the lower bound construction. Note that the portals all reside on the outer face of the embedding (see Figure 5). We also observe that the terminals of the flower gadget can be covered by a single face, namely the carpel of the flower.

We can define the direction of an edge both in Γ and Γ_t the following way: pq is a right edge if the right endpoint of p and q are equal, otherwise pq is a left edge. An edge $e = pq$ in Γ_t is isomorphic to an edge $e' = p'q'$ in Γ if $|p| = |p'|, |q| = |q'|$, and pq and $p'q'$ are both right or both left edges. It is easy to see that given a subtree S of Γ_t, there is an isomorphic tree S' in Γ. For this purpose, we define an isomorphism $\phi : V(S) \to V(\Gamma)$. Pick an arbitrary vertex $p \in V(S)$, and let $\phi(p) := p'$ where $p' \in V(\Gamma)$ is an arbitrary discrete interval for which $|p| = |p'|$. Using a depth-first search traversal of S, the picture of each vertex $v \in V(S)$ is uniquely defined: upon stepping from v to w in S, the size of $|w|$ compared to $|v|$ and the direction of the edge vw uniquely identifies $\phi(w)$ (given $\phi(v)$). Note that we do not run into conflicts (ϕ is injective), since any cycle in the image would imply the existence of a cycle in S, but S is a tree. For a tree S in Γ_t, fix an isomorphism ϕ_S, and let S_Γ be the image of the tree. Then we say that the *terminal sequence of* S is $[\![a]\!], \ldots, [\![b]\!]$ if this is the left-to right sequence of vertices in $V(S_\Gamma \cap H_0)$. Consequently, Lemma 4.1 can also be applied in Γ_t in the following sense. For a tree S with terminal sequence $[\![a]\!], \ldots, [\![b]\!]$ that induces p, its weight is at least $2(b-a) + \mathrm{dist}_\Gamma(\Delta([\![a, b]\!]), \phi(p))$.

The key theorem for using the flower gadget is the following.

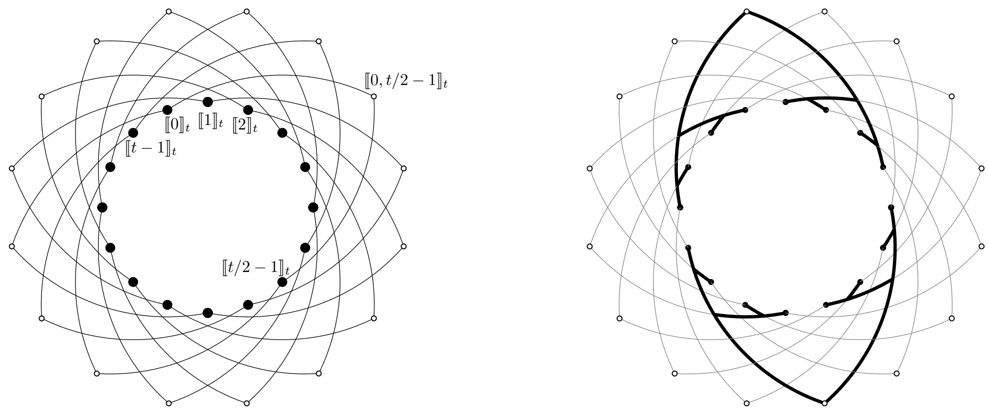

Figure 5: Left: The flower gadget, with the terminals as black disks and portals as circles. Right: A canonical solution.

THEOREM 4.3. *Let S be a Steiner forest in the flower gadget (with terminal set $[\![0]\!]_t, \dots [\![t-1]\!]_t$) where all trees of S contain a portal vertex. Then S has weight at least $2t-4$, and it can have weight exactly $2t-4$ only if it has at least two connected components, each component contains exactly one portal, and for any sequence of $t/2$ consecutive portals, at least one component has its portal there. Finally, if S has exactly two components and weight exactly $2t-4$, then S is* canonical, *that is, it induces exactly two opposite portal vertices: $[\![a, a+t/2-1]\!]_t$ and $[\![a+t/2, a-1]\!]_t$ for some $a \in \{1, \dots, t/2\}$.*

Proof. If S has only one component, then let p be a portal vertex induced by S. By Lemma 4.1, the tree has weight at least $2t-2$, which is strictly larger than $2t-4$. Hence, $w(S) > 2t-4$ or S has at least two connected components.

Now suppose that there are $k \geq 2$ components; we denote the i-th tree of S by S_i, let p_i be a portal vertex from S_i. We claim that the set of terminals corresponding to each tree must form a contiguous interval along the terminal face. To see this, suppose for contradiction that $[\![a]\!]_t, [\![b]\!]_t \in V(S_i)$ and $x \in [\![a, b]\!]_t \cap V(S_j)$ for some $i \neq j$. Then the shortest path in S_i from $[\![a]\!]_t$ to $[\![b]\!]_t$ in the planar embedding together with an arbitrary planar curve from $[\![a]\!]_t$ to $[\![b]\!]_t$ inside the terminal face forms a closed curve separating p_j and $[\![x]\!]_t$, or has p_j on its boundary and $[\![x]\!]_t$ inside. Therefore S_j cannot be disjoint from S_i. The claim follows.

Let $\phi_i = \phi_{S_i}$, and let $[\![a_i]\!] \dots [\![b_i]\!] \in V(\Gamma)$ be the terminal sequence of S_i. If we apply Lemma 4.1 for a tree S_i, we get

$$w(S_i) \geq 2(b_i - a_i + \mathrm{dist}(\phi_i(p_i), \Delta([\![a_i, b_i]\!])).$$

Let $\ell_i = |[\![a_i, b_i]\!]| - 1$. Observe that the terminals are always mapped into H_0 and the portal vertex p_i is always mapped into $H_{t/2-1}$ by all ϕ_i.

Consider the case when the triangle $\Delta([\![a_i, b_i]\!])$ does not reach $H_{t/2-1}$, so $\ell_i < t/2-1$. Then the distance $\mathrm{dist}(\phi_i(p_i), \Delta([\![a_i, b_i]\!]))$ is at least as big as the distance from the portal set $H_{t/2-1}$ to H_{ℓ_i} (Note that H_{ℓ_i} passes through $[\![a_i, b_i]\!]$.) The weight of the edges below $H_{t/2-1}$ is precisely $4/t$, and the number of edges required on a shortest path from $H_{t/2-1}$ to H_ℓ for some $\ell \leq t/2-1$ is $t/2-1-\ell$, so we have that

$$\mathrm{dist}(\phi_i(p_i), \Delta([\![a_i, b_i]\!])) \geq (t/2 - 1 - \ell_i)\frac{4}{t}. \tag{4.3}$$

If the triangle $\Delta([\![a_i, b_i]\!])$ reaches $H_{t/2}$, that is, if $\ell_i > t/2-1$, then inequality (4.3) still holds because the distance is nonnegative and the right hand side is nonpositive. Therefore by applying Lemma 4.1 to each component S_i and then applying the inequality (4.3) we get the following:

$$\begin{aligned} w(S) &= \sum_{i=1}^{k} w(S_i) \\ &\geq \sum_{i=1}^{k} \Big(2(|[\![a_i, b_i]\!]| - 1) + \mathrm{dist}(\phi_i(p_i), \Delta([\![a_i, b_i]\!]))\Big) \\ &\geq 2\sum_{i=1}^{k} \ell_i + \sum_{i=1}^{k} (t/2 - 1 - \ell_i) \cdot \frac{4}{t} \\ &= 2(t-k) + (kt/2 - k - (t-k)) \cdot \frac{4}{t} \\ &= 2t - 4. \end{aligned}$$

This shows that any Steiner forest of Γ_t where each tree contains a portal has weight at least $2t-4$. In case $w(S) = 2t-4$, both inequalities in this chain must be equalities, and in particular, each component attains equality for Lemma 4.1.

Suppose that $w(S) = 2t - 4$ and that a component S_i contains at least two portal vertices, p_i and p_i'. If we remove an edge from the tree path that goes from p_i to p_i', then all terminals remain connected to either p_i or p_i'. Therefore, we get a Steiner forest that is strictly lighter than $2t-4$ where all components contain a portal; this is contradiction.

Suppose that $w(S) = 2t - 4$ and there are $t/2$ consecutive portals not induced by any component of S; without loss of generality, suppose that these are $\llbracket 0, t/2-1\rrbracket_t, \llbracket 1, t/2\rrbracket_t \dots \llbracket t/2-1, t-2\rrbracket_t$. Let S_i be the tree in S that induces $\llbracket t/2-1\rrbracket_t$. Then $\phi_i(\llbracket t/2-1\rrbracket_t) \subset \llbracket a_i, b_i\rrbracket$. In the second inequality above, equality is only attainable if $\ell_j \le t/2-1$ for all j, since otherwise the distance can be lower bounded by 0 instead of the negative value we are using. Consequently, $\ell_i \le t/2-1$. Then in order for the equality to hold for S_i, it is necessary that $\phi_i(p_i)$ is at distance $(t/2-1-\ell_i)\frac{4}{t}$ from $\Delta(\llbracket a_i, b_i\rrbracket)$, which is only possible if $\llbracket a_i, b_i\rrbracket \subseteq \phi_i(p_i)$ by Lemma 4.1. Consequently, $\phi_i(\llbracket t/2-1\rrbracket_t) \subset \phi_i(p_i)$. Therefore, $p_i \in \{\llbracket 0, t/2-1\rrbracket_t, \llbracket 1, t/2\rrbracket_t \dots \llbracket t/2-1, t-2\rrbracket_t\}$.

Suppose that $w(S) = 2t-4$ and there are exactly two components in S. We want to show that S is canonical. If $\ell_1 \ne \ell_2$, then at least one of them is strictly larger than $t/2-1$; suppose that $\ell_1 > t/2-1$. Then in the above calculation we can lower bound $\mathrm{dist}(\phi_1(p_1), \Delta(\llbracket a_1, b_1\rrbracket))$ with 0 instead of $(t/2-1-\ell_1)\cdot\frac{4}{t}$, which yields a lower bound strictly larger than $2t-4$. Therefore, $\ell_1 = \ell_2 = t/2-1$, and the triangles $\phi_1^{-1}(\Delta(\llbracket a_1, b_1\rrbracket))$ and $\phi_2^{-1}(\Delta(\llbracket a_2, b_2\rrbracket_t))$ are completely contained in the flower gadget, with their tip being two opposite portal vertices $\llbracket a_1, b_1\rrbracket_t = \llbracket a_1, a_1+t/2-1\rrbracket_t$ and $\llbracket a_2, b_2\rrbracket_t = \llbracket a_1+t/2, a_1-1\rrbracket_t$. These are the unique portal vertices in S_1 and S_2 respectively. ☐

4.2 Verification Gadgets We first construct a verification gadget VG_N. We use exactly the same gadget as Marx et al. (see Figure 6). [3] The gadget VG_N has $2N+1$ so-called *portals*, which will be identified with or connected to portals of other gadgets. To be precise, the gadget has

- portals $y[1], \dots, y[N], w, z[N], \dots, z[1]$, which appear in this order along the outer face of VG_N;
- vertices $v[i,j]$ for each $i, j \in [N]$;
- edges from $y[i]$ to $v[1,i]$ of weight iM_2 and from $z[i]$ to $v[N,i]$ of weight iM_3 for each $i \in [N]$;

[3] Figures 6 and 7 from [34] were reproduced here with an author's permission.

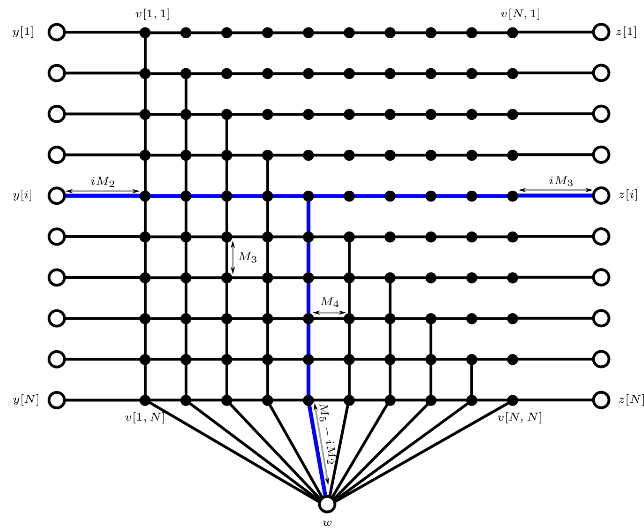

Figure 6: The verification gadget VG_N from [34, Figure 9]. The open circles indicate the portals that are connected to other parts of the graph. The blue edges indicate the connected subgraph mentioned in Lemma 4.2(i).

- 'horizontal' edges from $v[i,j]$ to $v[i+1,j]$ of weight M_4 for each $i \in [N-1]$ and $j \in [N]$;
- 'vertical' edges from $v[i,j]$ to $v[i,j+1]$ of weight M_3 for each $i \in [N]$ and $j \in [N] \setminus [i-1]$;
- edges from $v[i,N]$ to w of weight $M_5 - iM_2$ for each $i \in [N]$.

We call the edge between $v[i,N]$ to w the i-selector of VG_N. We actually require the so-called S-reduction of VG_N, denoted VG_N^S, for a set $S \subseteq [N]$, which is obtained from VG_N by removing the edges from $v[i,N]$ to w for each $i \notin S$.

The following lemma summarizes the properties we require of this gadget.

Lemma 4.2. *Let $S \subseteq [N]$. Then*

(i) for any $i \in [S]$, there is a connected subgraph of VG_N^S of weight $M_5 + (N-1)M_4 + (N-1)M_3$ that contains $y[i]$, $z[i]$, w, and the i-selector;

(ii) any connected subgraph H of VG_N^S that contains $y[i]$, $z[j]$, and w for $i, j \in [N]$ has weight at least $M_5 + (N-1)M_4 + (N-1)M_3$; moreover, if H has weight less than $M_5 + (N-1)M_4 + (N-1)M_3 + M_2$, then $i = j$ and H contains the i-selector and no other selector edge;

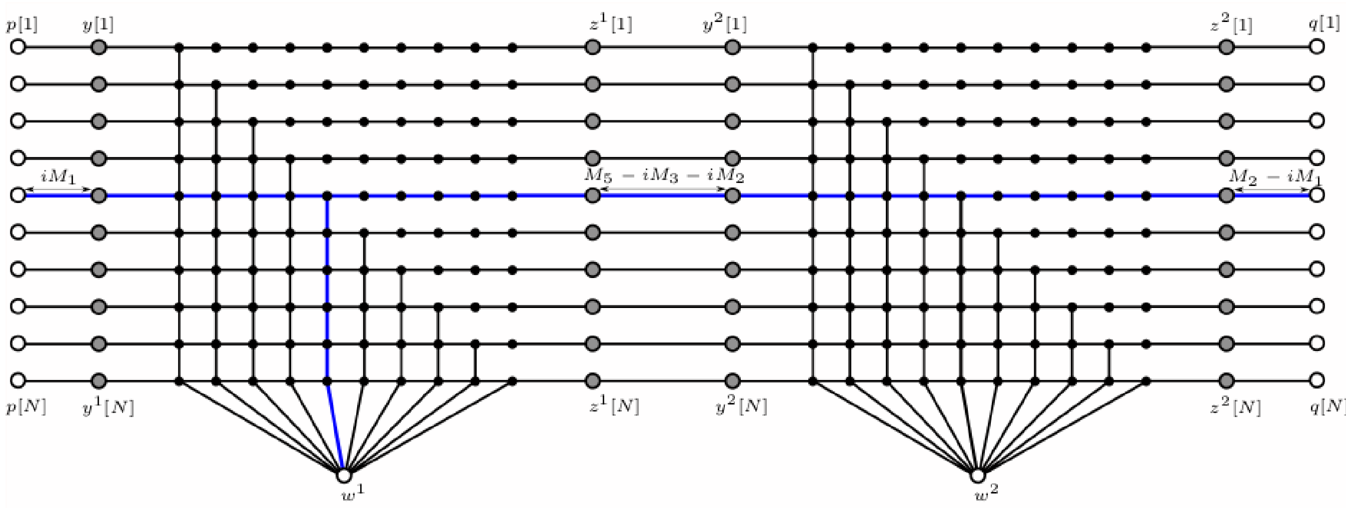

Figure 7: The paired verification gadget from [34, Figure 9]. In our gadget L-VG_N, the edges incident to w^i are omitted depending on S_i

(iii) there is a connected subgraph of VG_N^S *of weight* $(N-1)M_4 + iM_2 + iM_3$ *that contains* $y[i]$ *and* $z[i]$ *for* $i \in [N]$*;*

(iv) any connected subgraph of VG_N^S *that contains* $y[i]$ *and* $z[j]$ *for* $i, j \in [N]$ *has weight at least* $(N-1)M_4 + iM_2 + iM_3 + 2 \cdot \max\{0, j-i\} \cdot M_3$.

Since most properties are similar to those derived in [34, Lemma 7.5, 7.6], we defer the proof of this lemma to the full version.

Marx et al. then pair two verification gadgets, see Figure 7. We generalize their construction to combine L verification gadgets. To be precise, our gadget L-VG_N has

- portal vertices $p[1], \dots, p[N], w[1], \dots, w[L]$, and $q[N], \dots, q[1]$, which appear in this order along the outer face of VG_N;
- L verification gadgets VG_N. Let $y^\ell[1], \dots, y^\ell[N], w^\ell$, and $z^\ell[N], \dots, z^\ell[1]$ denote the portals of $\mathsf{VG}_N^{S_i}$ and identify $w[\ell]$ with w^ℓ;
- edges from $p[i]$ to $y^1[i]$ of weight iM_1 for $i \in [N]$;
- edges from $q[i]$ to $z^L[i]$ of weight $M_2 - iM_1$ for $i \in [N]$;
- edges e_i^ℓ from $z^\ell[i]$ to $y^{\ell+1}[i]$ of weight $M_5 - iM_3 - iM_2$ for $i \in [N]$ and $\ell \in [L-1]$, called the connector edges.

For $L = 2$, this is exactly the paired verification gadget of [34]. We require the $\mathcal{S}$-reduction of L-VG_N, which for $\mathcal{S} = \{S_1, \dots, S_L\}$ where $S_\ell \subseteq [N]$ for $\ell \in [L]$, contains the S_ℓ-reduction $\mathsf{VG}_N^{S_\ell}$ as the ℓ-th verification gadget (instead of the plain vanilla VG_N) for $\ell \in [L]$.

The following lemma summarizes the properties we require of this gadget.

LEMMA 4.3. *Let* $\mathcal{S} = \{S_1, \dots, S_L\}$*, where* $S_\ell \subseteq [N]$ *for* $\ell \in [L]$*. Then*

(i) for any $\ell \in [L]$ *and* $i \in S_\ell$*, there is a connected subgraph of* L-$\mathsf{VG}_N^{\mathcal{S}}$ *of weight* $LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2$ *that contains* $p[i]$*,* $q[i]$*,* $w[\ell]$ *and the* i*-selector incident on* $w[\ell]$*;*

(ii) any connected subgraph H *of* L-$\mathsf{VG}_N^{\mathcal{S}}$ *that contains* $p[i]$*,* $q[j]$*, and* $w[\ell]$ *for some* $i, j \in [N]$ *and* $\ell \in [L]$ *has weight at least* $LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2$*; moreover, if* H *has exactly this weight, then* $i = j$ *and* H *contains exactly one selector edge, namely the* i*-selector incident on* $w[\ell]$*.*

Since most properties are similar to those derived in [34, Lemma 7.5, 7.6], we defer the proof of this lemma to the full version.

4.3 Construction In the next two subsections, we aim to prove the following theorem, which will quickly imply Theorem 4.2.

THEOREM 4.4. *Let* $\mathcal{M}$ *be an instance of* GRID TILING*, with associated integers* n *and* k*. Then in time polynomial in* n *and* k*, one can construct an integer* $K_{\mathcal{M}}$ *and* *a planar graph $G_{\mathcal{M}}$ with positive edge weights and set $T_{\mathcal{M}}$ of terminals such that*

- *$G_{\mathcal{M}}$ has size $O(k^2n^5)$;*
- *$T_{\mathcal{M}}$ can be covered by $k(k-1)+1$ faces of $G_{\mathcal{M}}$;*
- *each edge has weight $O(k^{14}n^{22})$;*
- *$\mathcal{M}$ admits a solution if and only if $G_{\mathcal{M}}$ admits a Steiner tree of weight at most $K_{\mathcal{M}}$.*

Consider an instance $\mathcal{M}$ of Grid Tiling consisting of two integers n and k, and k^2 sets $M_{a,b} \subseteq [n] \times [n]$ for $a, b \in [k]$. By increasing n if necessary, we may assume that n is a power of 2. Throughout, let $N = n^2$, $L = n$, $t = 2L = 2n$, and $M = 10k^2NL$; observe that $M > 10NL$ as required. Figure 8 is provided to get a better understanding of the construction.

For each $a, b \in [k]$, we create two gadgets $G^W_{a,b} = L\text{-}\mathsf{VG}_N^{\mathcal{S}^W_{a,b}}$ and $G^E_{a,b} = L\text{-}\mathsf{VG}_N^{\mathcal{S}^E_{a,b}}$ for well-chosen sets $\mathcal{S}^W_{a,b}$ and $\mathcal{S}^E_{a,b}$. The w-portals of $G^W_{a,b}$ will face south, while the w-portals of $G^E_{a,b}$ will face north (i.e. $G^E_{a,b}$ is $L\text{-}\mathsf{VG}_N^{\mathcal{S}^E_{a,b}}$ rotated by 180 degrees). The q-portals of both gadgets will then be connected: we place an edge from $q^W[j]$ to $q^E[N-j+1]$ for each $j \in [N]$. The idea will be that selecting a 'row' in $G^W_{a,b}$ and $G^E_{a,b}$ corresponds to selecting a valid choice $(x_a, y_b) \in M_{a,b}$. We then add a flower gadget between $\mathcal{S}^W_{a,b}$ and $\mathcal{S}^E_{a,b+1}$ to ensure that the same y_b is chosen in each column of the Grid Tiling instance, while a simpler construction ensures that the same x_a is chosen in $\mathcal{S}^W_{a+1,b}$. We now describe the construction in more detail.

Let $a, b \in [k]$. We aim to construct $\mathcal{S}^W_{a,b}$ and $\mathcal{S}^E_{a,b}$, so that $G^W_{a,b}$ and $G^E_{a,b}$ are well defined. For each $l \in [L]$, let $S^{W,l}_{a,b} = \{(i-1)n + l \mid (i,l) \in M_{a,b}\}$ and let $S^{E,l}_{a,b} = \{N - ((i-1)n+l)+1 \mid (i,l) \in M_{a,b}\}$. Then $\mathcal{S}^W_{a,b} = \{S^{W,1}_{a,b}, \ldots, S^{W,L}_{a,b}\}$ and $\mathcal{S}^E_{a,b} = \{S^{E,1}_{a,b}, \ldots, S^{E,L}_{a,b}\}$. Now let $G^W_{a,b} = L\text{-}\mathsf{VG}_N^{\mathcal{S}^W_{a,b}}$ and $G^E_{a,b} = L\text{-}\mathsf{VG}_N^{\mathcal{S}^E_{a,b}}$. We use $p^W_{a,b}[1], \ldots, p^W_{a,b}[N], w^W_{a,b}[1], \ldots, w^W_{a,b}[L], q^W_{a,b}[N], \ldots, q^W_{a,b}[1]$ to denote the portals of $G^W_{a,b}$ and $p^E_{a,b}[1], \ldots, p^E_{a,b}[N], w^E_{a,b}[1], \ldots, w^E_{a,b}[L], q^E_{a,b}[N], \ldots, q^E_{a,b}[1]$ to denote the portals of $G^E_{a,b}$. Now we connect $q^W_{a,b}[j]$ with $q^E_{a,b}[N-j+1]$ for each $j \in [N]$ by a join edge $e^j_{a,b}$ of weight M_6. Denote the resulting gadget by $G_{a,b}$.

We now fuse the gadgets $G_{a,b}$ for fixed $a \in [k]$. Let $b \in [k-1]$ and $i \in [n]$. Create a new vertex $f_{a,b}[i]$. For each $l \in [n]$, add an edge from $p^E_{a,b}[N-((i-1)n+l)+1]$ to $f_{a,b}[i]$ and from $f_{a,b}[i]$ to $p^W_{a,b+1}[(i-1)n+l]$, both of weight M_6. The idea of $f_{a,b}[i]$ is that it allows us to switch the value selected in our solution of the Grid Tiling instance between column b and column $b+1$, while the value selected for row a remains the same (namely i).

Let $b \in [k]$. For each $a \in [k-1]$, create a flower gadget $F_{a,b}$ of size t. Since n is a power of 2, this is indeed possible. Multiply all weights in the gadget by $2^{\log t}M_7 = tM_7$, so that each weight is at least M_7 and at most $O(LM_7)$. Identify the vertex $[\![l,\, l+t/2-1]\!]_t$ of the flower gadget with $w^W_{a,b}[l]$ and the vertex $[\![l+t/2,\, l-1]\!]_t$ of the flower gadget with $w^E_{a+1,b}[l]$. If the Steiner tree in $F_{a,b}$ is canonical, then we can ensure that the value selected in our solution of the Grid Tiling instance for column b is the same, namely l, throughout. Finally, create a single terminal vertex $F_{0,b}$ and identify it with $w^E_{a,b}[l]$ for all $l \in [L]$, and create a single terminal vertex $F_{k,b}$ and identify it with $w^W_{a,b}[l]$ for all $l \in [L]$. We call these the dummy terminals.

As a last step, create a terminal r and k terminals $h_1, \ldots, h_k$. Add an edge of weight M_6 from r to all vertices $p^W_{a,1}[j]$ for $a \in [k]$ and $j \in [N]$. For each $a \in [k]$ and all $j \in N$, add an edge of weight M_6 from $q^E_{a,k}[j]$ to h_a. For notational convenience, we will sometimes write that $f_{a,0}[i] = r$ and $f_{a,k}[i] = h_a$ for $a \in [k]$ and $i \in [n]$.

Finally, let $K_{\mathcal{M}} := k(k-1)\cdot(2t-4)\cdot tM_7 + 3k^2M_6 + 2k^2\cdot(LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2)$.

This completes the construction. Observe that the resulting graph $G_{\mathcal{M}}$ is planar. Moreover, $G_{\mathcal{M}}$ has exactly $k(k-1)+1$ faces that jointly contain all terminals: $k(k-1)$ faces that form the carpels of the flower gadgets, plus the outer face of $G_{\mathcal{M}}$. Finally, observe that $G_{\mathcal{M}}$ has $O(k^2N^2L + k^2L^2) = O(k^2n^5)$ vertices.

4.4 Correctness

Lemma 4.4. *If $\mathcal{M}$ admits a solution, then $G_{\mathcal{M}}$ admits a Steiner tree of weight at most*

$$K_{\mathcal{M}} = k(k-1)\cdot(2t-4)\cdot tM_7 + 3k^2M_6 + 2k^2\cdot(LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2).$$

Proof. Let $x_1, \ldots, x_k, y_1, \ldots, y_k$ be a solution. We construct a tree as follows. For each $a, b \in [k]$, it follows from Lemma 4.3(i) and the construction of $\mathcal{S}^W_{a,b}$ that $G^W_{a,b}$ has a connected subgraph of weight $LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2$ that contains $p^W_{a,b}[x_a\cdot n + y_b]$, $q^W_{a,b}[x_a\cdot n + y_b]$, and $w^W_{a,b}[y_b]$. Similarly, $G^E_{a,b}$ has a connected subgraph of weight $LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2$ that contains $p^E_{a,b}[(n-x_a+1)\cdot n + (n-y_b+1)]$, $q^E_{a,b}[(n-x_a+1)\cdot n + (n-y_b+1)]$, and $w^E_{a,b}[y_b]$. Since $q^W_{a,b}[x_a\cdot n + y_b]$ and $p^E_{a,b}[(n-x_a+1)\cdot n + (n-y_b+1)]$ are connected by a join edge, we obtain a connected subgraph $H_{a,b}$ that contains $p^W_{a,b}[x_a\cdot n + y_b]$ and $p^E_{a,b}[(n-x_a+1)\cdot n + (n-y_b+1)]$ of total weight

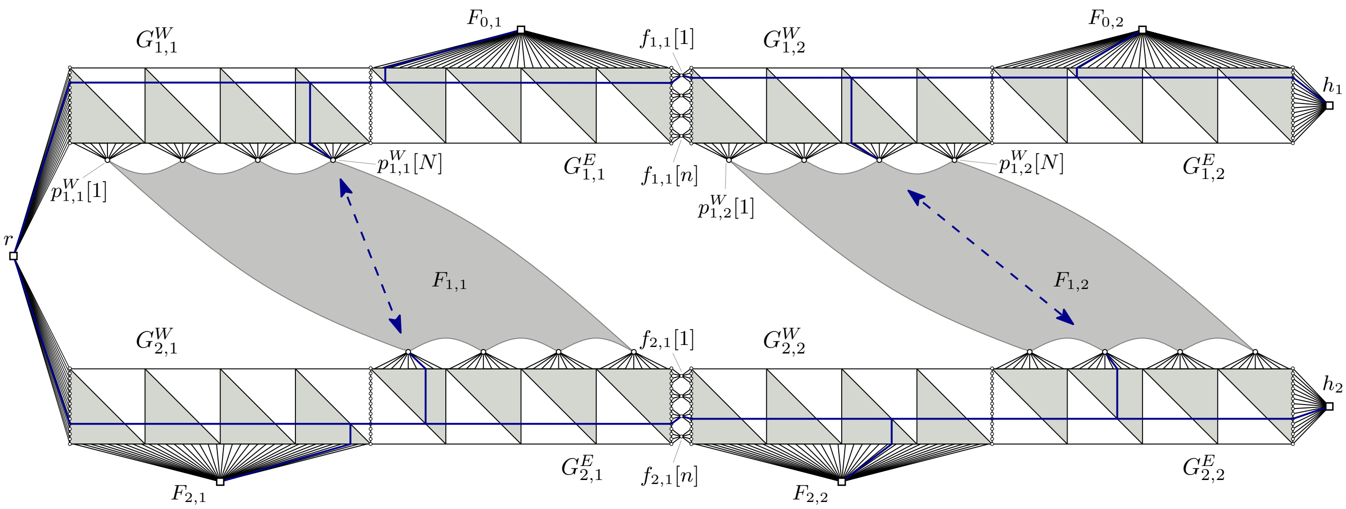

Figure 8: The graph $G_{\mathcal{M}}$. In both rows, two copies of L-VG_N are fused as described in Subsection 4.3. The gray clouds that connect the copies of L-VG_N vertically indicate flower gadget, and the blue arrows indicates the matched entry points into the flower gadget.

$M_6 + 2(LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2)$. Hence, the total weight of the union of the connected subgraphs $H_{a,b}$ over all $a, b \in [k]$ is $k^2M_6 + 2k^2 \cdot (LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2)$.

Let $b \in [k]$. For each $a \in [k-1]$, it follows from Theorem 4.3 that $F_{a,b}$ has a canonical Steiner forest $H^F_{a,b}$ on connector vertices $w^W_{a,b}[y_b] = [\![y_b,\, y_b + t/2 - 1]\!]_t$ and $w^E_{a+1,b}[y_b] = [\![y_b + t/2,\, y_b - 1]\!]_t$ of total weight $2t - 3 \cdot tM_7$. Hence, the total weight of the Steiner forests $H^F_{a,b}$ over all $a \in [k-1], b \in [k]$ is $k(k-1) \cdot (2t-4) \cdot tM_7$. Observe that $H^F_{a,b}$ has two connected components: the first is attached to $H_{a,b}$ through $w^W_{a,b}[y_b] = [\![y_b,\, y_b + t/2 - 1]\!]_t$; the second is attached to $H_{a+1,b}$ through $w^E_{a+1,b}[y_b] = [\![y_b + t/2,\, y_b - 1]\!]_t$.

Finally, for each $a \in [k], b \in [k-1]$, we select $f_{a,b}[x_a]$. Note that $f_{a,b}[x_a]$ is adjacent to $p^E_{a,b}[(n - x_a + 1) \cdot n + (n - y_b + 1)]$, and thus to $H_{a,b}$, through an edge of weight M_6. Similarly, $f_{a,b}[x_a]$ is adjacent to $p^W_{a,b+1}[x_a \cdot n + y_b]$, and thus to $H_{a,b+1}$, through an edge of weight M_6. Moreover, for each $a \in [k]$, r is adjacent to $p^W_{a,1}[x_a \cdot n + y_1]$, and thus to $H_{a,1}$, through an edge of weight M_6. Similarly, h_a is adjacent to $p^E_{a,k}[(n - x_a + 1) \cdot n + (n - y_k + 1)]$, and thus to $H_{a,k}$, through an edge of weight M_6. Let H denote the resulting subgraph.

Observe that H is connected by construction. Moreover, the weight of H is

$$\begin{aligned} &k(k-1) \cdot (2t-4) \cdot tM_7 + 3k^2M_6 \\ &+ 2k^2 \cdot (LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2). \end{aligned}$$

Furthermore, H contains all terminals, including the dummy terminals. Hence, by taking a spanning tree of H, the lemma follows. □

Lemma 4.5. *If $G_{\mathcal{M}}$ admits a Steiner tree of weight at most*

$$\begin{aligned} K_{\mathcal{M}} = {}&k(k-1) \cdot (2t-4) \cdot tM_7 + 3k^2M_6 \\ &+ 2k^2 \cdot (LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2), \end{aligned}$$

then $\mathcal{M}$ admits a solution.

Proof. Let H be the assumed Steiner tree; without loss of generality, H is inclusion-wise minimal. Let H_F denote the restriction of H to the flower gadgets, and $H_{F_{a,b}}$ the restriction of H to $F_{a,b}$. In general, H_F is a forest. We observe that

$$M_7 > 3k^2M_6 + 2k^2 \cdot (LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2)$$

by the choice of M. Since every edge in the flower gadgets has weight at least M_7 and is a multiple of M_7, it follows that H_F has weight at most $k(k-1) \cdot (2t-4) \cdot tM_7$. As any path from a terminal in a flower gadget to r contains a portal of that flower gadget, the minimality of H implies that all trees of H_F contain a portal of the corresponding gadget. In particular, for each $a \in [k-1], b \in [k]$, $H_{F_{a,b}}$ contains a portal of $F_{a,b}$. Then, by Theorem 4.3, it follows that $H_{F_{a,b}}$ has weight at least $(2t-4) \cdot tM_7$. Since there are $k(k-1)$ flower gadgets, this implies that H_F has weight at least $k(k-1) \cdot (2t-4) \cdot tM_7$, and thus weight exactly $k(k-1) \cdot (2t-4) \cdot tM_7$. Hence, for each $a \in [k-1], b \in [k]$, $H_{F_{a,b}}$ has weight exactly $(2t-4)\cdot tM_7$. Then Theorem 4.3 implies that each tree of $H_{F_{a,b}}$ (and in H_F) contains exactly one portal, and for any sequence of $t/2$ consecutive portals, at least one tree of $H_{F_{a,b}}$ has its portal there. Moreover, H has weight at most $3k^2M_6+2k^2\cdot(LM_5+L(N-1)M_4+(N-1)M_3+M_2)$ outside the flower gadgets.

Since each tree in H_F contains exactly one portal, the path P_a in H between r and h_a for $a \in [k]$ cannot cross a flower gadget and is fully contained in $\{r, h_a\} \cup \left(\bigcup_{b\in[k]} G_{a,b}\right) \cup \left(\bigcup_{b\in[k-1],i\in[n]} f_{a,b}[i]\right)$. In particular, the path contains at least $k-1$ fuse vertices and two edges of weight M_6 incident on each of them, at least k join edges of weight M_6, an edge of weight M_6 incident on r, an edge of weight M_6 incident on h_a. Hence, the path has weight at least $3kM_6$. Moreover, since P_a cannot cross the flower gadgets, the k paths $P_1, \ldots, P_k$ from r to $h_1, \ldots, h_k$ respectively are internally vertex disjoint. Hence, the total weight of the aforementioned edges across all of the k paths is $3k^2M_6$. Since $M_6 > 2k^2 \cdot (LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2)$ by the choice of M, it follows that H contains no further edges of weight M_6. In particular, H has weight at most $2k^2 \cdot (LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2)$ in total in the gadgets $G^W_{a,b}$ and $G^E_{a,b}$ for $a, b \in [k]$.

The preceding implies that H contains exactly $k(k-1)$ fuse vertices, one for each $a \in [k], b \in [k-1]$, denoted $f_{a,b}[i_{a,b}]$ for suitable $i_{a,b} \in [n]$. For notational convenience, define $i_{a,0} = i_{a,1}$ and $i_{a,k} = i_{a,k-1}$ for each $a \in [k]$. Moreover, each of these fuse vertices has degree exactly 2 in H. Similarly, $h_1, \ldots, h_k$ each have degree exactly 1 in H, and r has degree exactly k. Finally, H contains exactly one join edge $e^{j_{a,b}}_{a,b}$ for each $a, b \in [k]$ for certain $j_{a,b} \in [N]$. From this, we conclude that for each $a, b \in [k]$, H contains exactly one p-portal and exactly one q-portal of each of $G^W_{a,b}$ and $G^E_{a,b}$, specifically $p^W_{a,b}[\mu^W_{a,b}], q^W_{a,b}[\nu^W_{a,b}], p^E_{a,b}[\mu^E_{a,b}], q^E_{a,b}[\nu^E_{a,b}]$ for suitable $\mu^W_{a,b}, \nu^W_{a,b}, \mu^E_{a,b}, \nu^E_{a,b} \in [N]$.

Let $b \in [k]$ and $a \in [k-1]$. Since for any sequence of $t/2$ consecutive portals of $F_{a,b}$, at least one tree of $H_{F_{a,b}}$ has its portal there, it follows that one of $w^W_{a,b}[l_{a,b}] = [\![l_{a,b},\, l_{a,b}+t/2-1]\!]_t$ and one of $w^E_{a+1,b}[l'_{a,b}] = [\![l'_{a,b}+t/2,\, l'_{a,b}-1]\!]_t$ is in $H_{F_{a,b}}$ for certain $l_{a,b} \in [L]$ and $l'_{a,b} \in [L]$. From this and the placement of the dummy terminals, we conclude that for each $a, b \in [k]$, H contains at least one w-portal of each of $G^W_{a,b}$ and $G^E_{a,b}$, specifically $w^W_{a,b}[\lambda^W_{a,b}]$ and $w^E_{a,b}[\lambda^E_{a,b}]$ for suitable $\lambda^W_{a,b}, \lambda^E_{a,b} \in [L]$.

Now note that for each $a, b \in [k]$, H contains exactly one p-portal, exactly one q-portal, and at least one w-portal of each of $G^W_{a,b}$ and $G^E_{a,b}$. Moreover, H restricted to $G_{a,b}$ (denoted $H_{G_{a,b}}$) must be connected in order for a path from r to h_a to exist in H. Since only one join edge of $G_{a,b}$ is in H, as established previously, it follows that H restricted to $G^W_{a,b}$ (denoted $H_{G^W_{a,b}}$) and to $G^E_{a,b}$ (denoted $H_{G^E_{a,b}}$) must each be connected. Then Lemma 4.3(ii) implies that $H_{G^W_{a,b}}$ and $H_{G^E_{a,b}}$ each have weight at least $LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2$. Since H has weight at most $2k^2 \cdot (LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2)$ in total in the gadgets $G^W_{a,b}$ and $G^E_{a,b}$ for $a, b \in [k]$, it follows that $H_{G^W_{a,b}}$ and $H_{G^E_{a,b}}$ each have weight exactly $LM_5 + L(N-1)M_4 + (N-1)M_3 + M_2$ for each $a, b \in [k]$.

Let $a, b \in [k]$. Since $H_{G^W_{a,b}}$ has weight exactly $LM_5+L(N-1)M_4+(N-1)M_3+M_2$, is connected, and contains $p^W_{a,b}[\mu^W_{a,b}]$, $q^W_{a,b}[\nu^W_{a,b}]$, and $w^W_{a,b}[\lambda^W_{a,b}]$, it follows from Lemma 4.3(ii) that $\mu^W_{a,b} = \nu^W_{a,b}$ and that $H_{G^W_{a,b}}$ contains only one selector edge, namely the $\mu^W_{a,b}$-selector incident on $w^W_{a,b}[\lambda^W_{a,b}]$. Consequently, $w^W_{a,b}[\lambda^W_{a,b}]$ is the only portal among $w^W_{a,b}[1], \ldots, w^W_{a,b}[L]$ that is in $H_{G^W_{a,b}}$. Similar statements hold mutatis mutandis with respect to $H_{G^E_{a,b}}$. Since $H_{G_{a,b}}$ contains exactly one join edge, it follows that $\mu^W_{a,b} = \nu^W_{a,b} = N - \mu^E_{a,b} + 1 = N - \nu^E_{a,b} + 1$. The construction of $\mathcal{S}^W_{a,b}$ and $\mathcal{S}^E_{a,b}$ implies that $\lambda^W_{a,b} = \lambda^E_{a,b}$. Also, note that H must contain the edge between $f_{a,b-1}[i_{a,b-1}]$ and $p^W_{a,b}[\mu^W_{a,b}]$ as well as the edge between $f_{a,b}[i_{a,b}]$ and $p^E_{a,b}[\mu^E_{a,b}]$. The fact that $\mu^W_{a,b} = N - \mu^E_{a,b} + 1$ implies that $i_{a,b-1} = i_{a,b}$ by the definition of the fuse vertices. Hence, for each $a \in [k]$, it follows that $i_{a,0} = \cdots = i_{a,k}$. Set $x_a = i_{a,0}$ for each $a \in [k]$.

Let $a \in [k-1], b \in [k]$. By the preceding paragraph, $w^W_{a,b}[\lambda^W_{a,b}]$ is the only portal among $w^W_{a,b}[1], \ldots, w^W_{a,b}[L]$ that is in $H_{G^W_{a,b}}$, and $w^E_{a+1,b}[\lambda^E_{a+1,b}]$ is the only portal among $w^E_{a+1,b}[1], \ldots, w^E_{a+1,b}[L]$ that is in $H_{G^E_{a+1,b}}$. This implies that $H_{F_{a,b}}$ has exactly two components. We previously established that $H_{F_{a,b}}$ has weight exactly $(2t-4)\cdot tM_7$. Then Theorem 4.3 implies that $H_{F_{a,b}}$ is canonical, meaning that the two portals of $H_{F_{a,b}}$ in $F_{a,b}$ are opposite. By the construction of $G_{\mathcal{M}}$, this implies that $\lambda^W_{a,b} = \lambda^E_{a+1,b}$. Recall that $\lambda^W_{a,b} = \lambda^E_{a,b}$ and $\lambda^W_{a+1,b} = \lambda^E_{a+1,b}$ was established in the previous paragraph. Hence, for each $b \in [k]$, it follows that $\lambda^W_{1,b} = \cdot = \lambda^W_{k,b} = \lambda^E_{1,b} = \cdots = \lambda^E_{k,b}$. Set $y_b = \lambda^W_{1,b}$ for each $b \in [k]$.

We claim that $x_1, \ldots, x_k, y_1, \ldots, y_k$ is a solution to the Grid Tiling instance. Let $a, b \in [k]$. By definition, $\lambda^W_{a,b} = \lambda^E_{a,b} = y_b$ and $i_{a,b-1} = i_{a,b} = x_a$. Note that $\mu^W_{a,b} \in [\![(x_a-1)n+1, x_a n]\!]$ by the construction of $G_{\mathcal{M}}$ and by the fact that $p^W_{a,b}[\mu^W_{a,b}]$ is the only p-portal of $G^W_{a,b}$ in H. Since the $\mu^W_{a,b}$-selector incident on $w^W_{a,b}[\lambda^W_{a,b}] = w^W_{a,b}[y_b]$ is in H, $\mu^W_{a,b} = (x_a-1)n + y_b$.

The construction of $L\text{-}\mathsf{VG}_N^{\mathcal{S}}$ implies that $\mu_{a,b}^W \in S_{a,b}^{W,y_b}$. Then the construction of $\mathcal{S}_{a,b}^W$, and specifically of $S_{a,b}^{W,y_b}$ implies that $(x_a, y_b) \in M_{a,b}$. The claim follows, and thus so does the lemma. $\square$

The construction and the above lemmas immediately imply Theorem 4.4. The proof of Theorem 4.2 then quickly follows.

Proof of Theorem 4.2. Let $G_{\mathcal{M}}$ be the edge-weighted planar graph resulting from Theorem 4.4, with terminal set $T_{\mathcal{M}}$. Subdivide an edge e of $G_{\mathcal{M}}$ of weight $w > 1$ exactly $w-1$ times, such that e is replaced by a path of w unit-weight edges. Call the resulting graph $\mathcal{G}_{\mathcal{M}}$.

The bound on the size is immediate from the fact that $G_{\mathcal{M}}$ has $O(k^2n^5)$ edges of weight $O(k^{14}n^{22})$ each.

Note that there is a bijection between the faces of $\mathcal{G}_{\mathcal{M}}$ and of $G_{\mathcal{M}}$. Moreover, $T_{\mathcal{M}}$ is still present in $\mathcal{G}_{\mathcal{M}}$. Hence, the terminals of $T_{\mathcal{M}}$ can be covered by $k(k-1)+1$ faces of $\mathcal{G}_{\mathcal{M}}$.

The final property follows immediately from the subdivision of the edges in correspondence to their weights and from the corresponding property in Theorem 4.4. $\square$

5 Concluding Remarks

In this paper we gave an $2^{O(k)}n^{O(\sqrt{k})}$ time algorithm for Planar Steiner Tree, if the terminals are covered by k faces, and showed this is almost optimal assuming the Exponential Time Hypothesis. The crucial idea in the algorithm was to study seperators in a graph with artificially added edges that enforce how connected components are divided. The crucial idea in the lower bound is the flower gadget that is a graph with all terminals on one face where an optimal forest consisting of two trees can divide the terminal set arbitrarily in two parts.

Several exciting questions remain. First, an interesting question is whether our techniques could inspire further progress in any of the studies that invoked the original algorithm of Erickson [19]. For example in the mentioned approximation and kernelization algorithms [7, 39] the authors reduce the general Planar Steiner Tree to the case where terminals lie on one face. A natural direction to explore is to reduce the number of faces with terminals to more than one, and subsequently use the insights from this paper to aim for improved algorithms. It would also be interesting to see whether our techniques have consequences in the more geometric setting outlined by Provan [41, 42].

Second, a natural question is whether the $2^{O(k)}$ term in our running time can be removed. This would significantly generalize the $n^{O(|\sqrt{T}|)}W$-time algorithm of [34]. A natural approach would be to combine our technique with the technique of [34], but it seems highly unclear in which graph one should consider separators.

Acknowledgments

We thank Dániel Marx, Marcin Pilipczuk and Michał Pilipczuk for allowing us to use their figures.

References

[1] M. Bateni, C. Chekuri, A. Ene, M. Hajiaghayi, N. Korula, and D. Marx. Prize-collecting Steiner problems on planar graphs. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 1028–1049, 2011.

[2] M. Bateni, M. Hajiaghayi, and D. Marx. Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth. *J. ACM*, 58(5):21:1–21:37, 2011.

[3] M. W. Bern. *Network design problems: Steiner trees and spanning k-trees*. PhD thesis, University of California, Berkeley, 1987.

[4] M. W. Bern and D. Bienstock. Polynomially solvable special cases of the Steiner problem in planar networks. *Annals OR*, 33(6):403–418, 1991.

[5] D. Bienstock and C. L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM J. Comput.*, 17(1):53–76, 1988.

[6] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: Fast subset convolution. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 67–74, New York, NY, USA, 2007. ACM.

[7] G. Borradaile, C. Kenyon-Mathieu, and P. N. Klein. A polynomial-time approximation scheme for Steiner tree in planar graphs. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*, pages 1285–1294, 2007.

[8] G. Borradaile, P. N. Klein, and C. Mathieu. Steiner tree in planar graphs: An $O(n\log n)$ approximation scheme with singly-exponential dependence on epsilon. In *Algorithms and Data Structures, 10th International Workshop, WADS 2007, Proceedings*, pages 275–286, 2007.

[9] V. Bouchitté, F. Mazoit, and I. Todinca. Treewidth of planar graphs: connections with duality. *Electronic Notes in Discrete Mathematics*, 10:34 – 38, 2001. Comb01, Euroconference on Combinatorics, Graph Theory and Applications.

[10] D. Z. Chen and X. Wu. Efficient algorithms for k-terminal cuts on planar graphs. *Algorithmica*, 38(2):299–316, 2004.

[11] D. Z. Chen and J. Xu. Shortest path queries in planar graphs. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 469–478. ACM, 2000.

[12] S.-W. Cheng. *Exact Steiner Trees in Graphs and Grid Graphs*, pages 137–162. Springer US, Boston, MA, 2000.

[13] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[14] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs. *ACM Trans. Algorithms*, 1(1):33–47, July 2005.

[15] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

[16] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[17] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971.

[18] D. Du and X. Hu. *Steiner Tree Problems In Computer Communication Networks*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2008.

[19] R. E. Erickson, C. L. Monma, and A. F. Veinott Jr. Send-and-split method for minimum-concave-cost network flows. *Math. Oper. Res.*, 12(4):634–664, 1987.

[20] 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.

[21] 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*, pages 494–505, 2015.

[22] F. V. Fomin, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 515–524, 2016.

[23] L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404, 1954.

[24] G. N. Frederickson. Planar graph decomposition and all pairs shortest paths. *J. ACM*, 38(1):162–204, 1991.

[25] B. Fuchs, W. Kern, D. Mölle, S. Richter, P. Rossmanith, and X. Wang. Dynamic programming for minimum Steiner trees. *Theory Comput. Syst.*, 41(3):493–500, 2007.

[26] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*, volume 53 of *Annals of Discrete Mathematics*. Elsevier, 1992.

[27] R. Krauthgamer, J. R. Lee, and H. I. Rika. Flow-cut gaps and face covers in planar graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms SODA 2019*, 2019. To appear.

[28] R. Krauthgamer and I. Rika. Refined vertex sparsifiers of planar graphs. *CoRR*, abs/1702.05951, 2017.

[29] A. Y. Levin. Algorithm for the shortest connection of a group of graph vertices. *Soviet Mathematics Doklady*, 12:1477–1481, 1971.

[30] D. Lokshtanov and J. Nederlof. Saving space by algebraization. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010*, pages 321–330, 2010.

[31] B. Marshall. Faster exact algorithms for Steiner trees in planar networks. *Networks*, 20(1):109–120, 1990.

[32] D. Marx. A tight lower bound for planar multiway cut with fixed number of terminals. In A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 677–688. Springer, 2012.

[33] D. Marx and M. Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Proceedings*, pages 865–877, 2015.

[34] D. Marx, M. Pilipczuk, and M. Pilipczuk. On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science*, 2018. To appear. See also `https://arxiv.org/abs/1707.02190`.

[35] K. Matsumoto, T. Nishizeki, and N. Saito. An efficient algorithm for finding multicommodity flows in planar networks. *SIAM J. Comput.*, 14(2):289–302, 1985.

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

[37] H. Okamura and P. D. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory, Series B*, 31:75–81, 1981.

[38] 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*, pages 353–364, 2013.

[39] 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, 2014.

[40] H. Prömel and A. Steger. *The Steiner Tree Problem: A Tour through Graphs, Algorithms, and Complexity*. Advanced Lectures in Mathematics. Vieweg+Teubner Verlag, 2012.

[41] J. S. Provan. An approximation scheme for finding Steiner trees with obstacles. *SIAM J. Comput.*, 17(5):920–934, 1988.

[42] J. S. Provan. Convexity and the Steiner tree problem. *Networks*, 18(1):55–72, 1988.