Network Sparsification for Steiner Problems on Planar and Bounded-Genus Graphs*

Marcin Pilipczuk[†] Michał Pilipczuk[‡] Piotr Sankowski[§]
Erik Jan van Leeuwen[¶]

Abstract

We propose polynomial-time algorithms that sparsify planar and bounded-genus graphs while preserving optimal or near-optimal solutions to Steiner problems. Our main contribution is a polynomial-time algorithm that, given an unweighted graph G embedded on a surface of genus g and a designated face f bounded by a simple cycle of length k, uncovers a set $F \subseteq E(G)$ of size polynomial in g and k that contains an optimal Steiner tree for *any* set of terminals that is a subset of the vertices of f.

We apply this general theorem to prove that:

- given an unweighted graph G embedded on a surface of genus g and a terminal set $S \subseteq V(G)$, one can in polynomial time find a set $F \subseteq E(G)$ that contains an optimal Steiner tree T for S and that has size polynomial in g and $|E(T)|$;
- an analogous result holds for an optimal Steiner forest for a set $\mathcal{S}$ of terminal pairs;
- given an unweighted planar graph G and a terminal set $S \subseteq V(G)$, one can in polynomial time find a set $F \subseteq E(G)$ that contains an optimal (edge) multiway cut C separating S (i.e., a cutset that intersects any path with endpoints in different terminals from S) and that has size polynomial in $|C|$.

In the language of parameterized complexity, these results imply the first polynomial kernels for Steiner Tree and Steiner Forest on planar and bounded-genus graphs (parameterized by the size of the tree and forest, respectively) and for (Edge) Multiway Cut on planar graphs (parameterized by the size of the cutset). Steiner Tree and similar "subset" problems were identified in [Demaine, Hajiaghayi, Computer J., 2008] as important to the quest to widen the reach of the theory of bidimensionality ([Demaine et al., JACM 2005], [Fomin et al., SODA 2010]). Therefore, our results can be seen as a leap forward to achieve this broader goal.

Additionally, we obtain a weighted variant of our main contribution: a polynomial-time algorithm that, given an edge-weighted plane graph G, a designated face f bounded by a simple cycle of weight $w(f)$, and an accuracy parameter $\varepsilon > 0$, uncovers a set $F \subseteq E(G)$ of total weight at most $\mathrm{poly}(\varepsilon^{-1})w(f)$ that, for any set of terminal pairs that lie on f, contains

*An extended abstract of this work has appeared at FOCS 2014. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 267959 (Marcin Pilipczuk, Michał Pilipczuk), ERC Grant Agreement n. 259515 (Marcin Pilipczuk, Piotr Sankowski) and Foundation for Polish Science (Marcin Pilipczuk, Piotr Sankowski).

[†] University of Warwick, UK, `m.pilipczuk@dcs.warwick.ac.uk`. Research was done while the author was at University of Warwick, Poland, and University of Bergen, Norway.

[‡] Department of Informatics, University of Bergen, Norway, `michal.pilipczuk@ii.uib.no`. Research was done while the author was at University of Bergen, Norway.

[§] Institute of Informatics, University of Warsaw, Poland, `sank@mimuw.edu.pl`.

[¶] Max-Planck Institut für Informatik, Saarbrücken, Germany, `erikjan@mpi-inf.mpg.de`. Part of the research was done while the author was at Sapienza University of Rome, Italy.

a Steiner forest within additive error $\varepsilon w(f)$ from the optimal Steiner forest. This result deepens the understanding of the recent framework of approximation schemes for network design problems on planar graphs ([Klein, SICOMP 2008], [Borradaile, Klein, Mathieu, ACM TALG 2009], and later works) by explaining the structure of the solution space within a brick of the so-called *mortar graph*, the central notion of this framework.

1 Introduction

Preprocessing algorithms seek out and remove chunks of instances of hard problems that are irrelevant or easy to resolve. The strongest preprocessing algorithms reduce instances to the point that even an exponential-time brute-force algorithm can solve the remaining instance within limited time. The power of many preprocessing algorithms can be explained through the relatively recent framework of kernelization [29, 61]. In this framework, each problem instance I has an associated parameter $k(I)$, often the desired or optimal size of a solution to the instance. Then a *kernel* is a polynomial-time algorithm that preprocesses the instance so that its size shrinks to at most $g(k(I))$, for some computable function g. If g is a polynomial, then we call it a *polynomial kernel*.

The ability to measure the strength of a kernel through the function g has led to a concerted research effort to determine, for each problem, the function g of smallest order that can be attained by a kernel for it. Initial insight into this function, in particular a proof of its existence, is usually given by a *parameterized algorithm*: an algorithm that solves an instance I in time $g(k(I)) \cdot |I|^{O(1)}$. Such an algorithm implies a kernel with the same function g, while, if the considered problem is decidable, then any kernel immediately gives a parameterized algorithm as well [29, 61]. However, if the problem is NP-hard, then this approach can only yield a kernel of superpolynomial size, unless P=NP. Therefore, different insights are needed to find the function g of smallest order, and in particular to find a polynomial kernel. This fact, combined with the discovery that for many problems the existence of a polynomial kernel implies a collapse in the polynomial hierarchy [9, 40, 30], has recently led to a spike in research on polynomial kernels.

A focal point of research into polynomial kernels are problems on planar graphs. Many problems that on general graphs have no polynomial kernel or even no kernel at all, possess a polynomial kernel on planar graphs. The existence of almost all of these polynomial kernels can be explained from the theory of bidimensionality [10, 21, 39]. The core assumption behind this theory is that the considered problem is *bidimensional*: informally speaking, the solution to an instance must be dense in the input graph. However, this assumption clearly fails for a lot of problems, which has led to gaps in our understanding of the power of preprocessing algorithms for planar graphs. In their survey, Demaine and Hajiaghayi [22, 23] pointed out 'subset' problems, in particular Steiner Tree, as an important research goal in the quest to generalize the theory of bidimensionality.

In this paper, we pick up this line of research and positively resolve the question to the existence of a polynomial kernel on planar graphs for three well-known 'subset' problems: Steiner Tree, Steiner Forest, and Multiway Cut. We remark that the theory of bidimensionality does not apply to any of these three problems, and that for the first two problems a polynomial kernel on general graphs is unlikely to exist [27] and for the third the existence of a polynomial kernel on general graphs is a major open problem [19, 37, 55]. All kernelization results in this paper are a consequence of a single, generic sparsification algorithm for Steiner trees in planar graphs, which is of independent interest. This sparsification algorithm extends to edge-weighted planar graphs, and we demonstrate its impact on approximation algorithms for problems on planar graphs, in particular on the EPTAS for Steiner Tree on planar graphs [12].

1.1 Reading guide

The paper presents three views on our results, with increasing level of detail. In the first view, Section 1.2 states our results, and briefly describes our techniques and how they (vastly) differ from previous papers on planar graph problems . We discuss possible limitations and extensions in Section 1.3. The second view (in Section 2) provides a rich overview of the proofs of our results. Finally, the third view (Sections 3 through 14) gives full and detailed proofs.

1.2 Results

We present an overview of the three major results that make up this paper. First, we describe the generic sparsification algorithm for Steiner trees in planar graphs. Second, we show how this sparsification algorithm powers the kernelization results in this paper. Third, we exhibit the extension of the sparsification algorithm to edge-weighted planar graphs, and its implications for approximation algorithms on planar graphs.

The Main Theorem. In our main contribution, we characterize the behavior of Steiner trees in bricks. In our work, a *brick* is simply a connected plane graph B with one designated face formed by a simple cycle ∂B, which w.l.o.g. is the outer (infinite) face of the plane drawing of B, and called the *perimeter of* B. Recall that a *Steiner tree* of a graph G is a tree in G that contains a given set $S \subseteq V(G)$ (called *terminals*). We also say that the Steiner tree *connects* S. In the unweighted setting, a Steiner tree T that connects S is *optimal* if every Steiner tree that connects S has at least as many edges as T. We apply our characterization of Steiner trees in bricks to obtain the following sparsification algorithm:

Theorem 1.1 (Main Theorem)**.** *Let B be a brick. Then one can find in $\mathcal{O}(|\partial B|^{142} \cdot |V(B)|)$ time a subgraph H of B such that*

(i) $\partial B \subseteq H$,

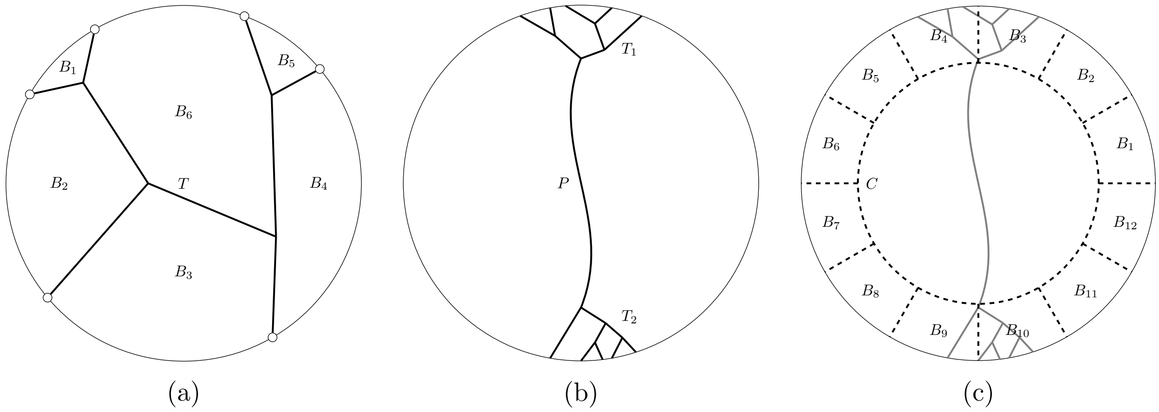
(ii) $|E(H)| = \mathcal{O}(|\partial B|^{142})$, and

(iii) for every set $S \subseteq V(\partial B)$, H contains some optimal Steiner tree in B that connects S.

The result of Theorem 1.1 is stronger than just a polynomial kernel, because the graph H contains an optimal Steiner tree for *any* terminal set that is a subset of the brick's perimeter. The result fits in a line of sparsification algorithms that reduce an instance and enable fast queries or computations (unknown at the current time) on the original instance, such as sparsification algorithms that approximately preserve vertex distances (so-called graph spanners) [62, 2], that preserve connectivity [59], or that conserve flows and cuts [43, 7, 6, 56]. Such sparsification algorithms are a common tool in, among others, dynamic graph algorithms [33], especially for planar graphs [34, 35, 26, 53, 66].

We also emphasise that the purely combinatorial (non-algorithmic) statement of Theorem 1.1, which asserts the existence of a subgraph H that has property (iii) and polynomial size, is nontrivial and, in our opinion, interesting on its own. A naive construction of a subgraph H that has property (iii) would mark an optimal Steiner tree for each set $S \subseteq V(\partial B)$. Combined with the observation that any optimal Steiner tree of a set $S \subseteq V(\partial B)$ has size at most $|\partial B|$ (as ∂B is a Steiner tree that connects S), we obtain a bound on the size of H of $|\partial B| \cdot 2^{|\partial B|}$. The polynomial bound of Theorem 1.1 presents a significant improvement over this naive bound.

The starting point of our work is the observation that an optimal Steiner tree for some choice of terminals on the perimeter decomposes the brick into smaller subbricks (see Figure 1a), on which we can subsequently recurse. However, the depth of the recursion may become too large if for *any* optimal Steiner tree for *any* choice of terminals on the perimeter, there are one or two subbricks that have perimeter almost equal to $|\partial B|$, as in Figure 1b. Therefore, the main part of our proof aims to understand the structure of the brick when this happens. In this case, we show that any optimal tree for any terminal set avoids a well-defined inside of the brick (the core), and give an algorithm to find it. Using the core and a deep topological analysis of the brick, we then find a cycle C of length $\mathcal{O}(|\partial B|)$ that lies close to the perimeter of B and that separates the core from all vertices of degree at least three of some optimal solution, for any set of terminals (see Figure 1c). Therefore, for any set of terminals, there is some optimal



(a) (b) (c)

Figure 1: (a) shows an optimal Steiner tree T and how it partitions the brick B into smaller bricks $B_1, \ldots, B_r$. (b) shows an optimal Steiner tree that connects a set of vertices on the perimeter of B and that consists of two small trees T_1, T_2 that are connected by a long path P; note that both bricks neighbouring P may have perimeter very close to $|\partial B|$. (c) shows a cycle C that (in particular) hides the small trees T_1, T_2 in the ring between C and ∂B, and a subsequent decomposition of B into smaller bricks.

solution whose intersection with the area inside C is a disjoint union of shortest paths, and thus we can sparsify this area by keeping a shortest path inside C between any pair of vertices of C. After this, we decompose the area between C and the perimeter of the brick into several smaller pieces, which we recursively sparsify. Using an inductive argument, we show this yields the polynomial bound on the size of the returned graph H.

We give a more detailed overview of the proof of Theorem 1.1 in Section 2. The full proof is contained in Sections 3–9. We also prove an analogue of Theorem 1.1 for graphs of bounded genus, with a polynomial dependence on the genus in the size bound. This analogue is sketched in Section 2.2 and presented in full in Section 12.

The approach that we take in this paper is very different from previous approaches to tackle problems on planar graphs or on bricks. In particular, our ideas are disjoint from those developed in both an EPTAS [12] and a subexponential-time parameterized algorithm [63] for Planar Steiner Tree. In those works, a brick was cut into so-called strips and then each strip was cut with a 'perpendicular column'. Therefore, already our starting observation (to use an optimal Steiner tree to decompose the brick) seems novel. Moreover, to the best of our knowledge, there is no work that aims to understand the behavior of a Steiner tree in a brick when all optimal Steiner trees leave one or two large subbricks (as in Figure 1b). Most of our paper is devoted to developing the tools and techniques to understand this case. We also stress that we do not employ any techniques used in the theory of bidimensionality. In particular, we do not use any tools from Graph Minors theory, such as the Excluded Grid Theorem [24, 65] — the engine of the theory of bidimensionality.

Applications of Theorem 1.1. We give three applications of Theorem 1.1. For each application, we state the result and its significance, and give an intuition of the proof. More detailed sketches of the proofs are provided in Section 2.3, and for details we refer to Section 11.

For the first application of Theorem 1.1, we consider Steiner Tree. For this problem, a polynomial kernel on general graphs implies a collapse of the polynomial hierarchy [27]. At the same time, the core assumption of bidimensionality theory fails, and whether a polynomial kernel exists for Steiner Tree on planar graphs was hitherto unknown. Using Theorem 1.1, we can resolve the existence of a polynomial kernel for Steiner Tree on planar graphs.

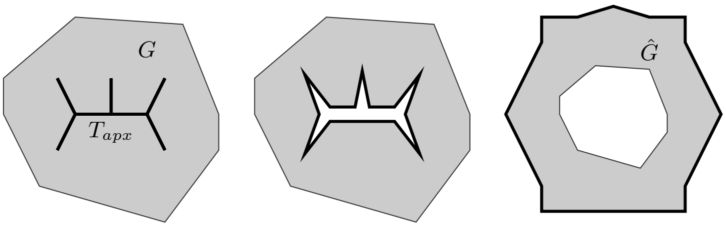


Figure 2: The process of cutting open the graph G along the approximate Steiner tree.

Theorem 1.2. *Given a* Planar Steiner Tree *instance* (G, S)*, one can in* $\mathcal{O}(k_{OPT}^{142}|G|)$ *time find a set* $F \subseteq E(G)$ *of* $\mathcal{O}(k_{OPT}^{142})$ *edges that contains an optimal Steiner tree connecting* S *in* G*, where* k_{OPT} *is the size of an optimal Steiner tree.*

We emphasise two aspects of Theorem 1.2. First, the proposed algorithm *does not* need to be given an optimal solution nor its size, even though the running time and output size of the algorithm are polynomial in the size of an optimal solution. Second, the running time of the algorithm can be bounded by $\mathcal{O}(|G|^2)$: if $|G|$ is smaller than the promised kernel bound, then the algorithm may simply return the input graph without any modification. Similar remarks hold also for the second and third applications of Theorem 1.1 that we present later.

Intuitively, Theorem 1.2 is almost a direct consequence of Theorem 1.1: we compute a 2-approximation to the optimal Steiner tree, cut the plane open along it, and then make the resulting cycle the outer face (see Figure 2), as done in the EPTAS for this problem [12]. Since all terminals lie on the outer face of the cut-open graph, we apply Theorem 1.1 to it, and project the resulting graph H back to the original graph.

For the second application of Theorem 1.1, we modify the approach of Theorem 1.2 for the closely related Steiner Forest problem on planar graphs. Recall that a *Steiner forest* that *connects* a family $\mathcal{S} \subseteq V(G) \times V(G)$ of terminal pairs in a graph G is a forest in G such that both vertices of each pair in $\mathcal{S}$ are contained in the same connected component of the forest.

Theorem 1.3. *Given a* Planar Steiner Forest *instance* $(G, \mathcal{S})$*, one can in* $\mathcal{O}(k_{OPT}^{710}|G|)$ *time find a set* $F \subseteq E(G)$ *of* $\mathcal{O}(k_{OPT}^{710})$ *edges that contains an optimal Steiner forest connecting* $\mathcal{S}$ *in* G*, where* k_{OPT} *is the size of an optimal Steiner forest.*

Using the analogue of Theorem 1.1 for bounded-genus graphs, we extend Theorems 1.2 and 1.3 to obtain a polynomial kernel for Steiner Tree and even Steiner Forest on such graphs (see Section 12). Here, we assume that we are given an embedding of the input graph into a surface of genus g such that the interior of each face is homeomorphic to an open disc.

For the third application of Theorem 1.1, we consider Edge Multiway Cut on planar graphs. Recall that an *edge multiway cut*[1] in a graph G is a set $X \subseteq E(G)$ such that no two vertices of a given set $S \subseteq V(G)$ are in the same component of $G \setminus X$. A recent breakthrough in the application of matroid theory to kernelization problems [54, 55] led to the discovery of a polynomial kernel for Multiway Cut on general graphs with a constant number of terminals. It is a major open question whether this problem has a polynomial kernel for an arbitrary number of terminals [19, 37, 55]. Here, we show that such a polynomial kernel does exist for Edge Multiway Cut on planar graphs.

[1] In the approximation algorithms literature, the term *multiway cut* usually refers to an edge cut, i.e., a subset of edges of the graph, and the node-deletion variants of the problem are often much harder. However, from the point of view of parameterized complexity, there is usually little or no difference between edge- and node-deletion variants of cut problems, and hence one often considers the (more general) node-deletion variant as the 'default one'. To avoid confusion, in this work we always explicitly state that we consider the edge-deletion variant.

Theorem 1.4. *Given a* Planar Edge Multiway Cut *instance (G, S), one can in polynomial time find a set $F \subseteq E(G)$ of $\mathcal{O}(k_{OPT}^{568})$ edges that contains an optimal solution to (G, S), where k_{OPT} is the size of this optimal solution.*

The proof of this theorem is based on a non-trivial relation between a multiway cut in a planar graph G and a Steiner tree in the dual of G. Hence, we apply Theorem 1.1 to a cut-open dual of the input graph. However, to bound the diameter of the initial brick, we need to bound the diameter of the dual of G. To this end, we show that edges are irrelevant to the problem if they are 'far' from some carefully chosen, laminar family of minimal cuts. Such edges may then be contracted safely, leading to the needed bound.

We note that in contrast to the work on polynomial kernels for Multiway Cut mentioned before [54, 55], we do not rely on matroid theory.

As an immediate consequence of Theorem 1.2 and Theorem 1.4, we observe that by plugging the kernels promised by these theorems into the algorithms of Tazari [67] for Planar Steiner Tree or its modification for Planar Edge Multiway Cut (provided for completeness in Section 13), or the algorithm of Klein and Marx [52] for Planar Edge Multiway Cut, respectively, we obtain faster parameterized algorithms for both problems.

Corollary 1.5. *Given a planar graph G, a terminal set $S \subseteq V(G)$, and an integer k, one can*

1. *in $2^{\mathcal{O}(\sqrt{k \log k})} + \mathcal{O}(k^{142}|V(G)|)$ time decide whether the* Planar Steiner Tree *instance (G, S) has a solution with at most k edges;*

2. *in $2^{\mathcal{O}(\sqrt{k} \log k)} + \mathrm{poly}(|V(G)|)$ time decide whether the* Planar Edge Multiway Cut *instance (G, S) has a solution with at most k edges;*

3. *in $2^{\mathcal{O}(|S| + \sqrt{|S|} \log k)} + \mathrm{poly}(|V(G)|)$ time decide whether the* Planar Edge Multiway Cut *instance (G, S) has a solution with at most k edges.*

This corollary improves on the subexponential-time algorithm for Planar Steiner Tree previously proposed by the authors [63], and on the algorithm for Planar Edge Multiway Cut by Klein and Marx [52] if $k = o(\log |V(G)|)$. As Tazari's algorithm extends to graphs of bounded genus, combining it with our kernelization algorithm, we obtain the first subexponential-time algorithm for Steiner Tree on graphs of bounded genus. The running time is a computable function of the genus times the running time of the planar case — see Corollary 12.5.

We also remark that a similar corollary is unlikely to exist for the case of Planar Steiner Forest. In Section 14 we observe that the lower bound for Steiner Forest on graphs of bounded treewidth of Bateni et al. [5], with minor modifications, shows also that Planar Steiner Forest does not admit a subexponential-time algorithm unless the Exponential Time Hypothesis of Impagliazzo, Paturi, and Zane [46] fails.

Theorem 1.6. *Unless the Exponential Time Hypothesis fails, no algorithm can decide in $2^{o(k)}\mathrm{poly}(|G|)$ time whether* Planar Steiner Forest *instances $(G, \mathcal{S})$ have a solution with at most k edges.*

Edge-Weighted Planar Graphs. Although the decomposition methods in the proof of Theorem 1.1 were developed with applications in unweighted graphs in mind, they can be modified for graphs with positive edge weights (henceforth called *edge-weighted graphs*). That is, we show the following weighted and approximate variant of Theorem 1.1:

Theorem 1.7. *Let $\varepsilon > 0$ be a fixed accuracy parameter, and let B be an edge-weighted brick. Then one can find in $\mathrm{poly}(\varepsilon^{-1})\,|B|\log|B|$ time a subgraph H of B such that*[2]

(i) $\partial B \subseteq H$,

(ii) $w(H) \leq \mathrm{poly}(\varepsilon^{-1})w(\partial B)$, *and*

(iii) for every set $\mathcal{S} \subseteq V(\partial B) \times V(\partial B)$ there exists a Steiner forest F_H that connects $\mathcal{S}$ in H such that $w(F_H) \leq w(F_B) + \varepsilon w(\partial B)$ for any Steiner forest F_B that connects $\mathcal{S}$ in B.

Notice that, contrary to Theorem 1.1, we state Theorem 1.7 in the language of Steiner *forest*, not Steiner tree. The reason is that the allowed error in Theorem 1.7 is additive, and therefore the forest statement seems significantly stronger than the tree one. Observe that for Theorem 1.1, it would be of no consequence to state it in the language of Steiner forest instead of in the language of Steiner tree.

The proof of Theorem 1.7 extends the techniques developed for Theorem 1.1 to edge-weighted planar graphs, and then wraps this extension into the mortar graph framework developed by Borradaille, Klein, and Mathieu [12]. Therefore, the main leap to prove Theorem 1.7 turns out to be a slight variant of Theorem 1.7, where $\mathcal{S}$ is allowed to contain at most θ terminal pairs and the obtained bound for $w(H)$ depends polynomially both on ε^{-1} and θ. We call this the θ-variant of Theorem 1.7.

The proof of this θ-variant considers a base case where $\mathcal{S}$ consists of a single terminal pair and only a $(1 + \varepsilon)$ multiplicative error in the weight of the forest F_H is allowed. Here, we first partition the brick B into so-called *strips* (as in [51]), and then provide an explicit construction of the graph H in a single strip. This base case, together with all the structural results and decomposition methods developed in the proof of Theorem 1.1 (extended to edge-weighted planar graphs), then powers the proof of the θ-variant of Theorem 1.7. We provide a more detailed sketch of the proof of Theorem 1.7 in Section 2.4, and a full proof in Section 10.

Theorem 1.7 influences the known polynomial-time approximation schemes for network design as follows. The mortar graph framework of Borradaille, Klein, and Mathieu [12] may be understood as a method to decompose a brick into cells, such that each cell is equipped with θ evenly-spaced portal vertices, and there is an approximate Steiner tree that for each cell uses a subset of the portal vertices to enter and leave the cell. Then it suffices to preserve an approximate or optimal Steiner tree for any subset of portal vertices. Previously, only a bound that is exponential in θ on the preserved subgraph of each cell was known [12]. The impact of our work, and particularly of the θ-variant of Theorem 1.7, is that the dependency on θ can be reduced to a polynomial. This observation is not only used to prove Theorem 1.7, but also leads to deeper understanding of the mortar graph framework.

Observe that one can directly derive an EPTAS for Planar Steiner Tree from Theorem 1.7: cut the input graph G open along a 2-approximate Steiner tree (as in the kernel, see Figure 2), apply Theorem 1.7 to the resulting brick B, and project the obtained graph H back onto the original graph. An optimal Steiner tree in G becomes an optimal Steiner forest in B, and thus the projection of H preserves an approximate Steiner tree for the input instance. Since the total weight of H is within a multiplicative factor $\mathrm{poly}(\varepsilon^{-1})$ of the weight of the optimal solution for the input instance, an application of Baker's shifting technique [3] can find an approximate solution in H in $2^{\mathrm{poly}(\varepsilon^{-1})}|H|\log|H|$ time. However, we note that the polynomial dependency on ε in the exponent is worse than the one obtained by the currently known EPTAS [12], despite our substantially improved reduction of the cells. This is because that EPTAS utilizes Baker's technique in a more clever way that is aware of the properties of the mortar graph, and is indifferent to the actual replacement within each cell.

[2] In this paper, we denote by $w(H)$ the total weight of all the edges of a graph H.

1.3 Discussion

A drawback of our methods is that the exponents in the kernel bounds and the polynomial dependency on ε^{-1} in the weighted variant are currently large, making the results theoretical. However, we see the strength of our results in that we prove that a polynomial kernel actually exists — thus proving that Planar Steiner Tree, Planar Steiner Forest, and Planar Edge Multiway Cut belong to the class of problems that have a polynomial kernel — rather than in the actual size bound. We believe that, using the road paved by our work, it is possible to decrease the exponent in the bound of the kernel. In fact, we conjecture that the correct dependency in Theorem 1.1 is quadratic, with a grid being the worst-case scenario.

Another limitation of our methods is that we need to parameterize by the *number of edges* of the Steiner tree. Although we believe that we can extend our results to kernelize Planar Steiner Tree with respect to the parameter *number of non-terminal vertices of the tree*, extending to the parameter *number of terminals* seems challenging. On general graphs, this parameter has already been studied, and it is known that the problem has a (tight) fixed-parameter algorithm [60, 18] and no polynomial kernel unless part of the polynomial hierarchy collapses [27]. However, our methods seem far from resolving whether Planar Steiner Tree has a polynomial kernel with respect to this smaller parameter. Similarly, one may consider graph-separation problems with vertex-based parameters, such as Odd Cycle Transversal or the node-deletion variant of Multiway Cut. On planar graphs, both of these problems are some sort of Steiner problem on the dual graph. It would be interesting to show polynomial kernels for these problems (without using the matroid framework [55]).

To generalize our methods, it would be interesting to lift our results to more general graph classes, such as graphs with a fixed excluded minor. For Edge Multiway Cut, even the bounded-genus case remains open. Further work is also needed to improve the allowed error in Theorem 1.7. Currently, this error is an additive error of $\varepsilon w(\partial B)$. In other words, a near-optimal Steiner forest is preserved only for "large" optimal forests, that is, for ones of size comparable to the perimeter of B. Is it possible to improve Theorem 1.7 to ensure a $(1+\varepsilon)$ multiplicative error? That is, to obtain a variant of Theorem 1.7 where the graph H satisfies $w(F_H) \leq (1+\varepsilon)w(F_B)$, and thus to preserve near-optimal Steiner forests at *all scales*? Finally, since our methods handle problems that are beyond the reach of the theory of bidimensionality, our contribution might open the door to a more general framework that is capable of addressing a broader range of problems.

1.4 Related work

The three problems considered in this paper (Steiner Tree, Steiner Forest, and Edge Multiway Cut) are all NP-hard [49, 20] and unlikely to have a PTAS [8, 20] on general graphs. However, they do admit constant-factor approximation algorithms [15, 1, 48].

Steiner Tree has a $2^{|S|}\cdot\text{poly}(|G|)$-time, polynomial-space algorithm on general graphs [60]; the exponential factor is believed to be optimal [18], but an improvement has not yet been ruled out under the Strong Exponential Time Hypothesis. The algorithm for Steiner Tree implies a $(2|\mathcal{S}|)^{|\mathcal{S}|}\cdot\text{poly}(|G|)$-time, polynomial-space algorithm for Steiner Forest. On the other hand, Edge Multiway Cut remains NP-hard on general graphs even when $|S| = 3$ [20], while for the parameterization by the size of the cut k, a $1.84^k\cdot\text{poly}(|G|)$-time algorithm is known [16].

Neither Steiner Tree nor Steiner Forest admits a polynomial kernel on general graphs [27], unless the polynomial hierarchy collapses. Recently, a polynomial kernel was given for Edge and Node Multiway cut for a constant number of terminals or deletable terminals [55]; nevertheless, the question for a polynomial kernel in the general case remains open.

Steiner Tree, Steiner Forest, and Edge Multiway Cut all remain NP-hard on planar graphs [41, 20], even in restricted cases. All three problems do admit an EPTAS on planar graphs [12, 32, 4], and Steiner Tree admits an EPTAS on bounded-genus graphs [11]. As mentioned before, for many graph problems on planar graphs, both polynomial kernels and subexponential-time algorithms follow from the theory of bidimensionality [21, 39]. However, the theory neither applies to Steiner Tree, Steiner Forest, nor Edge Multiway Cut.

We are not aware of any previous kernelization results for Steiner Tree, Steiner Forest, or Edge Multiway Cut on planar graphs. The question of the existence of a subexponential-time algorithm for Planar Steiner Tree was first explicitly pursued by Tazari [67]. He showed that such a result would be implied by a subexponential or polynomial kernel. The current authors adapted the main ideas of the EPTAS for Planar Steiner Tree [12] to show a subexponential-time algorithm [63], without actually giving a kernel beforehand. The algorithm of [63] in fact finds subexponentially many subgraphs of subexponential size, one of which is a subexponential kernel if the instance is a YES-instance. Finally, for Edge Multiway Cut on planar graphs, a $2^{\mathcal{O}(|S|)} \cdot |G|^{\mathcal{O}(\sqrt{|S|})}$-time algorithm is known [52] and believed to be optimal [58].

2 Overviews of the proofs

In this section, we give a more detailed overview of the proof of Theorem 1.1, its weighted counterpart Theorem 1.7, and discuss its corollaries (Theorems 1.2, 1.3 and 1.4).

Before we start, we set up some notation. For a subgraph H of G, we silently identify H with the edge set of H; that is, all our subgraphs are edge-induced. For a brick B, $\partial B[a, b]$ denotes the subpath of ∂B obtained by traversing ∂B in counter-clockwise direction from a to b. By Π we denote the standard Euclidean plane. For a closed curve γ on Π, we say that γ *strictly encloses* $c \in \Pi$ if $c \notin \gamma$ and γ is not continuously retractable in $\Pi \setminus \{c\}$ to a single point, and γ *encloses* c if it strictly encloses c or $c \in \gamma$. This notion naturally translates to cycles and walks in a plane graph G (strictly) enclosing vertices, edges, and faces of G.

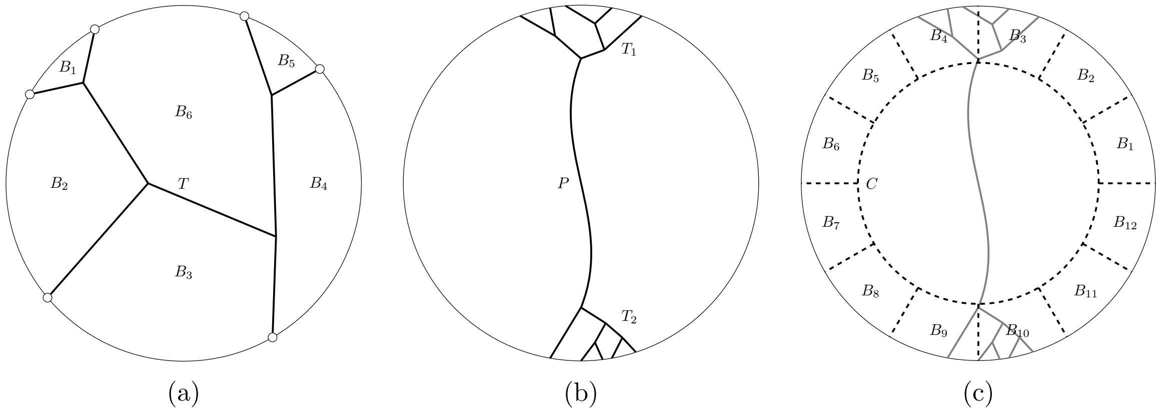
2.1 Overview of the proof of Theorem 1.1

The idea behind the proof of Theorem 1.1 is to apply it recursively on subbricks (subgraphs enclosed by a simple cycle) of the given brick B. The main challenge is to devise an appropriate way to decompose B into subbricks, so that their "measure" decreases. Here we use the *perimeter* of a brick as a potential that measures the progress of the algorithm.

Intuitively, we would want to do the following. Let T be a tree in B that connects a subset of the vertices on the perimeter of B. Then T splits B into a number of smaller bricks $B_1, \ldots, B_r$, formed by the finite faces of $\partial B \cup T$ (see Figure 3a). We recurse on bricks B_i, obtaining graphs $H_i \subseteq B_i$, and return $H := \bigcup_{i=1}^{r} H_i$. We can prove that this decomposition yields a polynomial bound on $|H|$ if (i) all bricks B_i have multiplicatively smaller perimeter than B, and (ii) the sum of the perimeters of the subbricks is linear in the perimeter of B.

In this approach, there are two clear issues that need to be solved. The first issue is that we need an algorithm to decide whether there is a tree T for which the induced set of subbricks satisfies conditions (i) and (ii). We design a dynamic programming algorithm that either correctly decides that no such tree exists, or finds a set of subbricks of B that satisfies condition (i) and (ii). In the latter case, we can recurse on each of those subbricks.

The second issue is that there might be no trees T for which the induced set of subbricks satisfies conditions (i) and (ii). In this case, optimal Steiner trees, which are a natural candidate for such partitioning trees T, behave in a specific way. For example, consider the tree of Figure 3b, which consists of two small trees T_1, T_2 that lie on opposite sides of the brick B and



(a) (b) (c)

Figure 3: (Figure 1 repeated); (a) shows an optimal Steiner tree T and how it partitions the brick B into smaller bricks $B_1, \ldots, B_r$. (b) shows an optimal Steiner tree that connects a set of vertices on the perimeter of B and that consists of two small trees T_1, T_2 that are connected by a long path P. (c) shows a cycle C that (in particular) hides the small trees T_1, T_2 in the ring between C and ∂B, and a subsequent decomposition of B into smaller bricks.

that are connected through a shortest path P (of length slightly less than $|\partial B|/2$). Then both faces of $\partial B \cup T$ that neighbour P may have perimeter almost equal to $|\partial B|$, thus blocking our default decomposition approach.

To address this second issue, we propose a completely different decomposition. Intuitively, we find a cycle C of length linear in $|\partial B|$ that lies close to ∂B, such that all vertices of degree three or more of any optimal Steiner tree are hidden in the ring between C and ∂B (see Figure 3c). We then decompose the ring between ∂B and C into a number of smaller bricks. We recursively apply Theorem 1.1 to these bricks, and return the result of these recursive calls together with a set of shortest paths inside C between any pair of vertices on C.

In Section 2.1.1 below, we formalise the above notions and give the algorithm that addresses the first issue. Then, Section 2.1.2 describes the default decomposition, whereas Section 2.1.3 describes the alternative decomposition that addresses the second issue. The full proof appears in Sections 3 through 9.

2.1.1 Deciding on the Decomposition

In this section, we present some of the basic notions of our paper and describe the algorithm that decides which of the two possible decompositions is used.

Definition 2.1. *For a brick B, a* brick covering *of B is a family $\mathcal{B} = \{B_1, \ldots, B_p\}$ of bricks, such that* (i) *each B_i, $1 \le i \le p$, is a subbrick of B, and* (ii) *each face of B is contained in at least one brick B_i, $1 \le i \le p$. A brick covering is called a* brick partition *if each face of B is contained in* exactly one *brick B_i.*

We note that if $\mathcal{B} = \{B_1, \ldots, B_p\}$ is a brick partition of B, then every edge of ∂B belongs to the perimeter of exactly one brick B_i, while every edge strictly enclosed by ∂B either is in the interior of exactly one brick B_i, or lies on the perimeters of exactly two bricks B_i, B_j for $i \neq j$.

Any connected set $F \subseteq B$ will be called a *connector*. Let S be the set of vertices of ∂B adjacent to at least one edge of F; the elements of S are the *anchors* of the connector F. We then say that F *connects* S. For a connector F, we say that F is *optimal* if there is no connector F' with $|F'| < |F|$ that connects a superset of the anchors of F. Clearly, each optimal connector F induces a tree, whose every leaf is an anchor of F. We say that a connector $F \subseteq B$ is *brickable* if the boundary of every inner face of $\partial B \cup F$ is a simple cycle, i.e., these boundaries form subbricks of B. Let $\mathcal{B}$ be the corresponding brick partition of B. Observe that $\sum_{B' \in \mathcal{B}} |\partial B'| \leq |\partial B| + 2|F|$.

Next, we define the crucial notions for partitions and coverings that are used for the default decomposition.

Definition 2.2. *The* total perimeter *of a brick covering $\mathcal{B} = \{B_1, \ldots, B_p\}$ is defined as $\sum_{i=1}^{p} |\partial B_i|$. For a constant $c > 0$, $\mathcal{B}$ is* c-short *if the total perimeter of $\mathcal{B}$ is at most $c \cdot |\partial B|$. For a constant $\tau > 0$, $\mathcal{B}$ is* τ-nice *if $|\partial B_i| \leq (1 - \tau) \cdot |\partial B|$ for each $1 \leq i \leq p$.*

Similarly, a brickable connector $F \subseteq B$, with $\mathcal{B} = \{B_1, \ldots, B_p\}$ being the corresponding brick partition, is c-short *if $\mathcal{B}$ is c-short, is simply* short *if it is 3-short, and is* τ-nice *if $\mathcal{B}$ is τ-nice.*

Observe that if $F \subseteq B$ is a brickable connector, then F is c-short if $|F| \leq |\partial B| \cdot (c - 1)/2$, and F is short if $|F| \leq |\partial B|$. Moreover, if F is an optimal connector, then F is a short brickable connector, as F must be a tree of length at most $|\partial B|$. Now we are ready to give the algorithm that decides what decomposition to use.

Theorem 2.3. *Let $\tau > 0$ be a fixed constant. Given a brick B, in $\mathcal{O}(|\partial B|^8 |B|)$ time one can either correctly conclude that no short τ-nice tree exists in B or find a 3-short τ-nice brick covering of B.*

The proof of Theorem 4.4, omitted in this overview and provided in full detail in Section 9, is a technical modification of the classical algorithm of Erickson et al. [36]. That algorithm computes an optimal Steiner tree in a planar graph assuming that all the terminals lie on the boundary of the infinite face. It uses the Dreyfus-Wagner dynamic-programming approach, where a state consists of a subset of already connected terminals, and the current "interface" vertex; the main observation is that only states with consecutive terminals on the boundary are relevant, yielding a polynomial bound on the number of them. In our case, we can proceed similarly: our state consists of the leftmost and rightmost chosen terminal, the "interface" vertex inside the brick, the total length of the tree, and the length of the leftmost and rightmost path in the constructed tree. Consequently, the terminals are chosen on-the-fly.

In case some short τ-nice tree exists, for technical reasons we cannot ensure that the output of the algorithm of Theorem 2.3 will actually be a brick partition corresponding to some short τ-nice tree. Instead, the algorithm may output a brick covering, but one that is guaranteed to be 3-short and τ-nice. This is sufficient for our purposes.

We can now formally describe the main line of reasoning of our sparsification algorithm. Let $\tau > 0$ be some constant chosen later. If $|\partial B| \leq 2/\tau$, then for each $S \subseteq V(\partial B)$ we compute an optimal Steiner tree that connects S using the algorithm of Erickson et al. [36], and take the union of all such trees. If $|\partial B| > 2/\tau$, then we run the algorithm of Theorem 2.3 for B and τ. If the algorithm returns a 3-short τ-nice brick covering, then we proceed to the default decomposition, formalized in Section 2.1.2 below. Otherwise, if the algorithm of Theorem 2.3 concluded that B does not contain any short τ-nice tree, then we proceed to the arguments in Section 2.1.3. We show that in all cases we obtain a subgraph of B that satisfies conditions (i)-(iii) of Theorem 1.1.

2.1.2 The Default Decomposition

Suppose that the algorithm of Theorem 2.3 returns a 3-short τ-nice brick covering $\mathcal{B} = \{B_1, \ldots, B_p\}$ of B. We can then use this brick covering as a decomposition and recurse on each brick individually. This is formalized in the following lemma.

Lemma 2.4. *Let $c, \tau > 0$ be constants. Let B be a brick and let $\mathcal{B} = \{B_1, \ldots, B_p\}$ be a c-short τ-nice brick covering of B. Assume that the algorithm of Theorem 1.1 was applied recursively to bricks $B_1, \ldots, B_p$, and let $H_1, \ldots, H_p$ be the subgraphs output by this algorithm for $B_1, \ldots, B_p$, respectively, where $|H_i| \le C \cdot |\partial B_i|^\alpha$ for some constants $C > 0$ and $\alpha \ge 1$ such that $(1-\tau)^{\alpha-1} \le \frac{1}{c}$. Let $H = \bigcup_{i=1}^{p} H_i$. Then H satisfies conditions (i)-(iii) of Theorem 1.1, with $|H| \le C \cdot |\partial B|^\alpha$.*

Proof. To see that H satisfies condition (i), note that every edge of ∂B is in the perimeter of some brick B_i, and that $\partial B_i \subseteq H_i$ for every $i = 1, 2, \ldots, p$. Therefore, $\partial B \subseteq H$.

To see that H satisfies condition (ii), recall that $\mathcal{B}$ is c-short and that $|\partial B_i| \le (1-\tau) \cdot |\partial B|$ for each $i = 1, 2, \ldots, p$. Therefore, $|\partial B_i|^\alpha \le |\partial B_i| \cdot (1-\tau)^{\alpha-1} |\partial B|^{\alpha-1}$, and

$$|H| \le \sum_{i=1}^{p} |H_i| \le C \cdot \sum_{i=1}^{p} |\partial B_i|^\alpha \le C \cdot (1-\tau)^{\alpha-1} |\partial B|^{\alpha-1} \cdot \sum_{i=1}^{p} |\partial B_i| \le c \cdot (1-\tau)^{\alpha-1} C \cdot |\partial B|^\alpha \le C \cdot |\partial B|^\alpha.$$

Finally, to see that H satisfies condition (iii), let $S \subseteq V(\partial B)$ be a set of terminals lying on the perimeter of B, and let T be an optimal Steiner tree connecting S in B that contains a minimum number of edges that are not in H. We claim that $T \subseteq H$. Assume the contrary, and let $e \in T \setminus H$. Since each face of B is contained in some brick of $\mathcal{B}$, there exists a brick B_i such that ∂B_i encloses e. As $\partial B_i \subseteq H_i \subseteq H$, we infer $e \notin \partial B_i$. Consider the subgraph of T strictly enclosed by ∂B_i, and let X be the connected component of this subgraph that contains e. Clearly, X is a connector inside B_i. Since H_i is obtained by a recursive application of Theorem 1.1, there exists a connected subgraph $D \subseteq H_i$ that connects the anchors of X and that satisfies $|D| \le |X|$. Let $T' = (T \setminus X) \cup D$. Observe that $|T'| \le |T|$ and that T' contains strictly less edges that are not in H than T does. Since D connects the anchors of X in H_i, T' still connects the anchors of T in B, that is, T' connects S. However, T is an optimal Steiner tree that connects S, and thus T' is also an optimal Steiner tree that connects S. Since T' contains strictly less edges that are not in H than T, this contradicts the choice of T. Hence, $T \subseteq H$. □

2.1.3 The Alternative Decomposition — Mountain Ranges and the Core

Suppose that the algorithm of Theorem 2.3 decides that no short τ-nice tree exists in B. As mentioned before, we want to find a cycle C of length linear in $|\partial B|$ that is close to ∂B, such that all vertices of degree three or more of any optimal Steiner tree are hidden in the ring between C and ∂B (see Figure 3c). In the following, we use a constant $\delta \in (0, \frac{1}{2})$, which depends on τ and is chosen later.

Definition 2.5. *A δ-*carve *L from a brick B is a pair (P, I), where P, called the* carvemark, *is a path in B between two distinct vertices $a, b \in V(\partial B)$ of length at most $(\frac{1}{2} - \delta) \cdot |\partial B|$, and I, called the* carvebase, *is a shortest of the two paths $\partial B[a, b], \partial B[b, a]$. The subgraph enclosed by the closed walk $P \cup I$ is called the* interior *of a δ-carve.*

Of particular interest will be the following special type of δ-carves.

Definition 2.6. *For fixed $l, r \in V(\partial B)$, a δ-*mountain *of B for l, r is a δ-carve M in B such that*

1. *l and r are the endpoints of the carvemark and carvebase of M;*

2. *the edges of the carvemark can be partitioned into two paths P_L, P_R, where P_L is a shortest l–P_R path in the interior of M and P_R is a shortest r–P_L path in the interior of M.*

We write $M = (P_L \wedge P_R)$ to exhibit the partition of the carvemark into paths P_L and P_R. We use v_M to denote the unique vertex of $V(P_L) \cap V(P_R)$. We also say that a δ-mountain M *connects* the vertices l and r.

The following lemma motivates why we are interested in δ-mountains. For a tree T, $T[a, b]$ denotes the unique path in T between vertices a and b.

Lemma 2.7. *Let B be a brick and let T be an optimal Steiner tree connecting $V(T) \cap V(\partial B)$ in B. Let $uv \in T$ be an edge of T, where v is of degree at least 3 in T, and let T_v be the connected component of $T \setminus \{uv\}$ containing v, rooted at v. Let l and r be the leftmost and rightmost elements of $V(T_v) \cap V(\partial B)$, that is, $V(T_v) \cap V(\partial B) \subseteq V(\partial B[l, r])$ and $T[l, r] \cup \partial B[r, l]$ encloses uv. Assume furthermore that $|\partial B[l, r]| < |\partial B|/2$. Then $M := (T[l, v] \wedge T[r, v])$ is a δ-mountain connecting l and r, for any $\delta < 1/2 - |T[l, r]|/|\partial B|$.*

Proof. As v is of degree at least 3 in T, v has degree at least 2 in T_v, and $T[l, v] \cap T[r, v] = \{v\}$. Therefore, $T[l, v] \cup T[v, r] = T[l, r]$, and $T[l, r]$ induces a δ-carve M with carvebase $\partial B[l, r]$.

Suppose that M is not a δ-mountain if we take $P_L = T[l, v]$ and $P_R = T[r, v]$. Without loss of generality, there exists a path P enclosed by M that connects l with $w \in V(P_R)$, $V(P_R) \cap V(P) = \{w\}$, and $|P| < |T[l, v]|$. Let D be the subgraph of M enclosed by the closed walk $T[l, v] \cup P \cup T[v, w]$. Define $T' := (T \setminus D) \cup T[v, w] \cup P$. As $T[v, w] \cup T[l, v] \subseteq D$, $|T'| < |T|$. By the definition of l and r, $T[l, v] \setminus P$ does not contain any vertex of ∂B. Therefore, T' is a connected subgraph of B connecting $V(T) \cap V(\partial B)$, a contradiction to the optimality of T. □

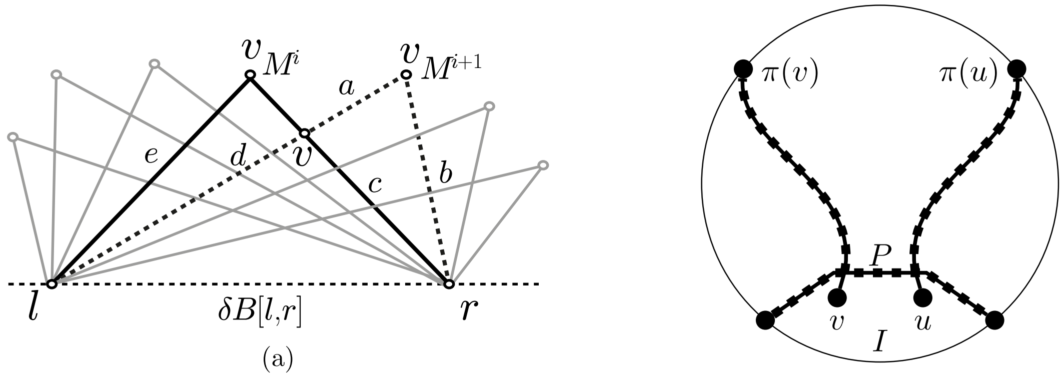
The above lemma shows that small subtrees of optimal Steiner trees in B are hidden in δ-mountains. Here, 'small' means that the leftmost and rightmost path in the subtree have total length at most $(1/2 - \delta) \cdot |\partial B|$. Note that an optimal Steiner tree in B has total size smaller than $|\partial B|$, as ∂B without an arbitrary edge connects any subset of $V(\partial B)$. Therefore, if we choose δ appropriately, then we can 'hide' almost an entire optimal non-τ-nice Steiner tree in at most two δ-mountains. To hide most of *all* optimal Steiner trees, we consider unions of δ-mountains. For fixed vertices $l, r \in V(\partial B)$, the *δ-mountain range* is the closed walk $W_{l,r}$ in B such that a face f of B is enclosed by $W_{l,r}$ if and only if f belongs to some δ-mountain that connects l and r.

Theorem 2.8 (Mountain Range Theorem)**.** *Fix $\tau \in [0, 1/4)$ and $\delta \in [2\tau, 1/2)$, and assume that B does not admit any short τ-nice tree. Then for any fixed $l, r \in V(\partial B)$ with $|\partial B[l, r]| < |\partial B|/2$, $W_{l,r}$ has length at most $3 \cdot |\partial B[l, r]|$. Moreover, the set of the faces enclosed by $W_{l,r}$ can be computed in $\mathcal{O}(|B|)$ time.*

Proof sketch. By case analysis, omitted in this overview and provided in full detail in Section 6, we deduce that the set of all inclusion-wise maximal δ-mountains essentially looks as in Figure 4a, i.e., for any two maximal mountains there exists exactly one region of the plane that is in one of them but not in the other one.

Let $\{M^i = (P_L^i, P_R^i)\}_{i=1}^s$ be the set of all these maximal δ-mountains, ordered from left to right. By induction, we show that the perimeter of the union of the first i δ-mountains, denoted p^i, is at most $|\partial B[l, r]| + |P_R^1| + |P_L^i|$. This statement clearly holds for $i = 1$, and for $i = s$ it proves the bound on the perimeter of the δ-mountain range promised by Theorem 2.8.

For the inductive step, define $b = |P_R^{i+1}|$ and $e = |P_L^i|$. Let v be the first point on P_L^{i+1} that lies on P_R^i. We denote the distance (along P_L^{i+1}) from l to v as d and the distance from v to $v_{M^{i+1}}$ as a. Finally, we denote by c the distance (along P_R^i) from r to v. These definitions are illustrated in Figure 4a. Observe that $d \geq e$, because M^i is a δ-mountain. Similarly, observe that $c \geq b$, because M^{i+1} is a δ-mountain. Hence, $p^{i+1} - p^i = a + b - c \leq a \leq a + d - e = |P_L^{i+1}| - |P_L^i|$. This concludes the inductive step. In this overview, we omit the description of the algorithm that finds the mountain range. □



(a)

(b)

Figure 4: (a) shows a mountain range. (b) shows a short τ-nice tree occurring if $\pi(v)$ and $\pi(u)$ are far from I.

We now designate $\mathcal{O}(\tau^{-1})$ vertices on ∂B, and construct the union $\mathcal{M}$ of all δ-mountain ranges for each pair of designated vertices. Using the following deep theorem, we can show that $\mathcal{M}$ is not the entire brick.

Theorem 2.9 (Core Theorem)**.** *For any $\tau \in (0, \frac{1}{4})$ and any $\delta \in [2\tau, \frac{1}{2})$, if B has no short τ-nice tree, then there exists a face of B that is not enclosed by any δ-carve. Moreover, such a face can be found in $O(|B|)$ time.*

Proof sketch. Suppose, for sake of contradiction, that all faces of B are enclosed by some δ-carve. We first observe that, for any brickable short tree T with diameter not more than $(\frac{1}{2} - \delta) \cdot |\partial B|$, there exists an interval I_T of ∂B of length at most $(\frac{1}{2} - \frac{\delta}{2}) \cdot |\partial B|$ such that all anchors of T are in I_T. If no such interval exists, then every brick induced by T has perimeter less than $(\frac{1}{2} - \delta) \cdot |\partial B| + (\frac{1}{2} + \frac{\delta}{2}) \cdot |\partial B| \leq (1 - \tau) \cdot |\partial B|$. Hence, T would be τ-nice, a contradiction.

Define a map $v \to \pi(v)$ for $v \in V(B)$ such that $\pi(v)$ is a vertex of ∂B closest to v. The main observation is that if v and u belong to the interior of some δ-carve (P, I), then the distance between $\pi(v)$ and $\pi(u)$ along ∂B is at most $(\frac{1}{2} - \frac{\delta}{2}) \cdot |\partial B|$. To see this, consider the shortest paths P_v from v to $\pi(v)$. These paths can be used to form a tree T, consisting of P, the subpath of P_v to $\pi(v)$ from the last point of P_v on P, and the subpath of P_u to $\pi(u)$ from the last point of P_u on P (see Figure 4b). We observe that the diameter of T is bounded by $|P| \leq (\frac{1}{2} - \delta) \cdot |\partial B|$, because the paths that make up T always have length at most the corresponding part of P. Moreover, as T has only four leaves, $|T|$ is bounded by twice the diameter of T, so T is short. Hence, $\pi(v)$, $\pi(u)$, and $V(P) \cap V(\partial B)$ lie on the interval I_T, as observed above. We extend π to the edges of B by mapping uv onto the shorter subpath between $\pi(u)$ and $\pi(v)$ on ∂B. Now consider a face f that is enclosed by (P, I). We note that no point of any edge of f is mapped to a point lying exactly opposite on ∂B to any point in $V(P) \cap V(\partial B)$, as such points cannot belong to I_T. Hence, all edges of f are mapped to an interval of ∂B. Since an interval is a simply connected metric space, we can extend π from the boundary of face f to its interior in a continuous manner such that the whole face f is mapped into it. Consequently, since every face of B can be enclosed by a δ-carve, we have constructed a retraction of a closed disc onto its boundary. This contradicts Borsuk's non-retraction theorem [13]. □

As each δ-mountain is actually a δ-carve, $\mathcal{M}$ does not contain an arbitrarily chosen core face $f_{\texttt{core}}$ promised by Theorem 2.9. Hence, the union of the perimeters of the δ-mountain ranges that make up $\mathcal{M}$ contains a cycle C_0 that separates $f_{\texttt{core}}$ from the mountain ranges. Moreover,

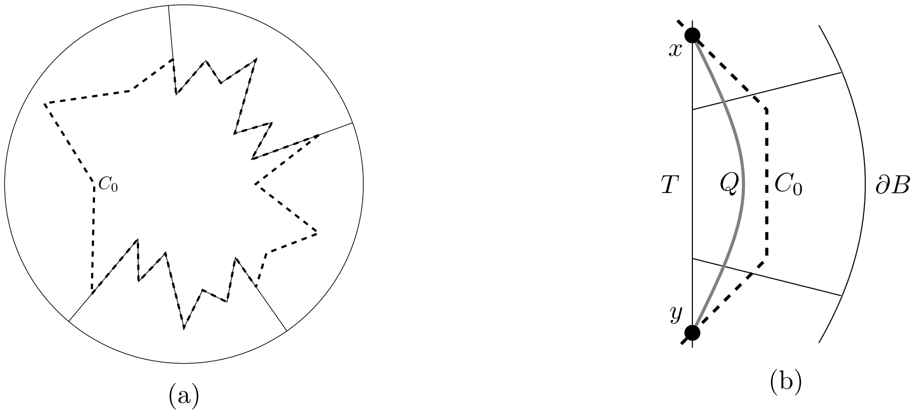


Figure 5: (a) shows the cycle C_0 formed by the union of the perimeters of the mountain ranges; example mountain ranges are drawn solid. (b) shows how to shortcut the tree T (solid) with a shortest xy-path Q (gray).

as we construct only $\mathcal{O}(\tau^{-2})$ mountain ranges, each of perimeter $\mathcal{O}(|\partial B|)$ by Theorem 2.8, we have that $|C_0| = \mathcal{O}(|\partial B|)$; see Figure 5a.

We observe that certain optimal Steiner trees in B may behave nontrivially in the subgraph enclosed by C_0, and in particular, may still have a vertex of degree three or more that is enclosed by C_0. However, this behavior is easily dealt with as follows. Consider the situation in Figure 5b. If Q is a shortest path between x and y, then we may replace the part of the tree to the left of Q by Q. Hence, we shortcut C_0 whenever possible while keeping $f_{\texttt{core}}$ enclosed by C_0. By choosing $\delta = 4\tau$, we then obtain the following result.

Theorem 2.10. *Let $\tau \in (0, 1/36]$. Assume that B does not admit any short τ-nice tree and that $|\partial B| > 2/\tau$. Then one can in $\mathcal{O}(|B|)$ time compute a simple cycle C in B with the following properties:*

1. *the length of C is at most $\frac{16}{\tau^2}|\partial B|$;*
2. *for each vertex $x \in V(C)$, there exists a path from x to $V(\partial B)$ of length at most $(\frac{1}{4} - 2\tau) \cdot |\partial B|$ such that no edge of the path is strictly enclosed by C;*
3. *C encloses $f_{\texttt{core}}$, where $f_{\texttt{core}}$ is a face of B, promised by Theorem 2.9, that is not enclosed by any 2τ-carve;*
4. *for any $S \subseteq V(\partial B)$, there exists an optimal Steiner tree T_S that connects S in B such that no vertex of degree at least 3 in T_S is strictly enclosed by C.*

Finally, we are ready to describe the decomposition. Apply the algorithm of Theorem 2.10 to B, and let C denote the resulting cycle. We can then decompose the brick as in Figure 3c, meaning that the area between C and ∂B is partitioned into a number of small subbricks of total perimeter $\mathcal{O}(|\partial B|)$. Here we use the second property of C that is promised by Theorem 2.10 to build the sides of the subbricks. We recursively apply Theorem 1.1 to these subbricks, and let H denote the union of the resulting subgraphs. Then we add to H for each pair of vertices of C a shortest path in the area enclosed by C between the two vertices if that shortest path has length at most $|\partial B|$. The linear bound on the total perimeter of the subbricks enables a similar analysis as in the proof of Lemma 2.4. We then choose $\tau = 1/36$. This concludes the proof of Theorem 1.1.

2.2 Extending to graphs of bounded genus

In this section, we informally argue how to extend Theorem 1.1 to graphs of bounded genus. A detailed statement of the result and a full proof can be found in Section 12.

We use the framework of Borradaile et al. [11]: the idea is to reduce the bounded-genus case to the planar case by cutting the graph embedded on a surface of bounded genus into a planar graph using a cutset of small size. That is, as in [11], given a brick embedded on a surface of genus g (i.e., a graph with a designated face), we may cut along a number of "short" cutpaths to make the brick planar, at the cost of extending the perimeter and the diameter of the brick by an additive factor of $\mathcal{O}(gd)$, where d is the diameter of G. However, in our case, d can be bounded by the perimeter of the brick, as vertices further from the perimeter may be safely discarded.

2.3 Applications of Theorem 1.1

In this section, we briefly sketch how to prove Theorems 1.2, 1.3, and 1.4. Full proofs appear in Section 11.

Proof sketch of Theorem 1.2. We manipulate the graph such that all terminals lie on the outer face. We first find a 2-approximate Steiner tree T_{apx} for S in G. We then cut the plane open along T_{apx}, cf. [12]. That is, we create an Euler tour of T_{apx} that traverses each edge twice in different directions and respects the plane embedding of T_{apx}. Then we duplicate every edge of T_{apx}, replace each vertex v of T_{apx} with $d-1$ copies of v, where d is the degree of v in T_{apx}, and distribute the copies in the plane embedding so that we obtain a new face F with boundary corresponding to the aforementioned Euler tour. Fix the embedding of the resulting graph $\hat{G}$ such that F is its outer face. Note that the terminals S lie only on the outer face of $\hat{G}$, and that $|\partial\hat{G}| \leq 4k_{OPT}$. Apply Theorem 1.1 to $\hat{G}$ to obtain $\hat{H}$, which is of size $\mathcal{O}(|\partial\hat{G}|^{142}) = \mathcal{O}(k_{OPT}^{142})$. As an optimal Steiner tree T in G splits into a family of trees in $\hat{G}$ that each connect subsets of $V(T) \cap V(\partial\hat{G})$, the projection of $\hat{H}$ onto G yields the desired set $F \subseteq E(G)$. □

To prove Theorem 1.3, we compute a simple approximate solution and remove all edges that are farther from a terminal than the size of this approximate solution. We then apply the same idea as in Theorem 1.2 to each of the resulting connected components.

The idea behind the proof of Theorem 1.4 is that the Edge Multiway Cut problem becomes a Steiner Forest-like problem in the dual graph. Hence, we cut open the dual of G similarly as we cut open G in Theorem 1.2: for each terminal t of G, we take the cycle C_t in the dual of G that consists of all edges incident to t, and cut the dual along a short connected subgraph containing all cycles C_t for all terminals of G. We show that to preserve an optimal solution for Edge Multiway Cut in G it suffices to preserve an optimal Steiner tree for any choice of the terminals on the perimeter of the obtained brick. Hence, to apply Theorem 1.1, we need to bound the length the perimeter, that is, the length of the subgraph of the dual of G that we cut along. By standard reductions, the total length of the cycles C_t (i.e., the total number of edges incident to terminals) is bounded by $2k_{OPT}$, where k_{OPT} is the optimal solution size. Hence, it suffices to bound the diameter of the dual of G.

To this end, we fix a terminal t and choose an inclusion-wise maximal laminar family of minimal separators that separate t from the remaining terminals and that are maximally "pushed away" from t (that is, they are important separators in the sense of [57]). By the "pushed away" property of the chosen family, each chosen separator is of different size, and as there are at most $2k_{OPT}$ edges incident to the terminals, the largest chosen separator is of size at most $2k_{OPT}$. Hence, there are $\mathcal{O}(k_{OPT}^2)$ edges in this chosen laminar family of minimal separators.

The essence of the proof is to show that an edge that is "far" from the chosen family of separators is irrelevant for the problem, and may be safely contracted. Here, "far" means ck_{OPT} for some universal constant c. Intuitively, if such an edge e is chosen in an optimal solution X, then the connected component of X of the dual of G that contains e lives between two separators from the chosen family, and we can show that it can be replaced by (a part of) one of these two separators.

Hence, after this reduction is performed exhaustively, the diameter of the dual of G is bounded by $\mathcal{O}(k_{OPT}^3)$. Consequently, cutting the graph open and applying Theorem 1.1 leads to a polynomial kernel.

Using the extension of Theorem 1.1 to graphs of bounded genus, we can extend Theorem 1.2 and 1.3, and part 1 of Corollary 1.5 to such graphs (see Section 12 for details).

2.4 The weighted variant: overview of the proof of Theorem 1.7

We now focus on the weighted variant, and sketch the proof of Theorem 1.7. A full proof appears in Section 10.

We start by considering a base case, where $\mathcal{S}$ consists of a single terminal pair and H must contain a Steiner forest F_H that connects $\mathcal{S}$ such that $w(F_H) \leq (1+\varepsilon)w(F_B)$ for any Steiner forest F_B in B that connects $\mathcal{S}$. To this end, we first partition the input brick into *strips* [51]. Informally speaking, a *strip* is a brick whose perimeter can be partitioned into a shortest path (called the *south boundary*) and an "almost" shortest path (called the *north boundary*). Note that any Steiner tree in a strip that cannot be replaced by a part of the perimeter, even with a small loss in weight, needs to contain terminals both on the south and north boundary. We use this observation to provide an explicit construction of the graph H in a single strip, using so-called columns (similar to the columns introduced in [12]).

With the base case of a single terminal pair in mind, we move to the θ-variant of Theorem 1.7, where $\mathcal{S}$ is allowed to contain only θ terminal pairs and the obtained bound for $w(H)$ depends polynomially both on ε^{-1} and θ. In this proof, we use the entire power of the structural results and decomposition methods developed for the proof of Theorem 1.1, adjusted to the edge-weighted case. In short, we show that if we decompose each brick recursively into smaller bricks, stopping when the perimeter of the brick drops below some threshold $\mathrm{poly}(\varepsilon/\theta)w(\partial B)$, then we can take the single-pair graph H developed previously in each such small brick, and the union of all such graphs has the desired properties. The crux of the analysis is that the bound θ ensures that we can "buy" the entire perimeter of each small brick in which some vertex of degree at least three of an optimal Steiner forest of B is present.

Finally, we use the partitioning methods from the EPTAS [12], the so-called mortar graph framework, to derive Theorem 1.7 from the θ-variant. The mortar graph constructed by [12] is essentially a brickable connector. We call the bricks induced by this connector *cells.* The mortar graph has the property that there exists a near-optimal Steiner forest in B that crosses each cell at most $\alpha(\varepsilon) = o(\varepsilon^{-5.5})$ times. Therefore, we construct the mortar graph of the input brick and then apply θ-variant to each cell independently, for an appropriate choice of $\theta = \mathrm{poly}(\varepsilon^{-1})$. This then yields the desired graph H.

3 Preliminaries

We use standard graph notation, see e.g. [25]. All our graphs are undirected and, unless otherwise stated, simple. For a graph G, by $V(G)$ and $E(G)$ we denote its vertex- and edge-set, respectively. For $v \in V(G)$, the neighbourhood of v is defined as $N_G(v) = \{u : uv \in E(G)\}$ and the closed neighbourhood of v as $N_G[v] = N_G(v) \cup \{v\}$. We extend these notions to sets

$X \subseteq V(G)$ as $N_G[X] = \bigcup_{v \in X} N_G[v]$ and $N_G(X) = N_G[X] \setminus X$. We omit the subscript if the graph is clear from the context.

For a subgraph H of G, we silently identify H with the edge set of H; that is, all our subgraphs are edge-induced. In particular, this applies to all paths, walks, and cycles; we treat them as sequences of edges.

In this paper, we work with both unweighted and edge-weighted graphs. An *edge-weighted graph* is a graph G equipped with a weight function $w : E(G) \to (0, +\infty)$. We explicitly disallow zero-cost edges in the input graph. For any edge $e \in E(G)$, the value $w(e)$ is the *length* or *weight* of the edge e. For any subgraph H of G (in particular, for any cycle or path in G), the *length* or *weight* of H is defined as $w(H) = \sum_{e \in H} w(e)$. An *unweighted graph* is an edge-weighted graph with weight function $w(e) = 1$ for each edge e, i.e., $w(H) = |H|$ for any subgraph H.

The distance between two vertices is the length of a shortest path between them. The distance between two vertex sets is the minimum distance between pairs of vertices in the sets. The distance between two (sets of) edges is the minimum distance between the endpoints of the edges. By $\mathrm{dist}_G(X, Y)$ we define the distance between objects (vertices, vertex sets, edge sets) X and Y in the graph G.

By Π we denote the standard euclidean plane. Let G be a plane graph, that is, a graph embedded on plane Π. Let γ be a closed curve on the plane, that is, a continuous image of a circle. We say that γ *strictly encloses* a point c on the plane if c does not lie on γ and γ is not the neutral element of the fundamental group of $\Pi \setminus \{c\}$ (note that this fundamental group is isomorphic to $\mathbb{Z}$) or, equivalently, c does not lie on γ and γ is not continuously retractable to a single point in $\Pi \setminus \{c\}$. We say that γ *encloses* c if γ strictly encloses c or c lies on γ. We often identify closed walks in the graph G with the closed curves that they induce in the planar embedding; thus, we can say that a closed walk in the graph (strictly) encloses c. We extend these notions to vertices, edges, and faces of a graph G: a vertex is (strictly) enclosed if its drawing on the plane is (strictly) enclosed, and edge is (strictly) enclosed if all interior points of its drawing are (strictly) enclosed, and a face is (strictly) enclosed if all points of its interior are (strictly) enclosed. We also say that a closed walk in G (strictly) encloses some object if the drawing of this closed walk (strictly) encloses the object. Note that if C is a simple cycle in G, then its drawing is a closed curve without self-intersections, and the notion of (strict) enclosure coincides with the intuitive meaning of these terms.

Definition 3.1. *A connected plane graph B is called a* brick *if the boundary of the infinite face of B is a simple cycle. This cycle is then called the* perimeter *of the brick, and denoted by ∂B. The* interior *of the brick, denoted* $\mathrm{int}B$, *is the graph induced by all the edges not lying on the perimeter, that is,* $\mathrm{int}B := B \setminus \partial B$.

Note that for a brick B, all the edges of $\mathrm{int}B$ as well as all the vertices of $\mathrm{int}B$ not lying on ∂B, are strictly enclosed by ∂B. For a brick B, every face of B enclosed by ∂B is called an *inner face*.

For a path P, we denote by $P[a, b]$ the subpath of P starting in vertex a and ending in vertex b. This definition is extended to the perimeter ∂B of a brick B in the following way. We denote by $\partial B[a, b]$ the subpath of ∂B obtained by traversing ∂B in counter-clockwise direction from a to b. On the other hand, for a tree T we denote by $T[a, b]$ the unique path in T between vertices a and b.

We also need the following notation. Let T be a tree embedded in the plane, and let uv be an edge of T. The *subtree of T, rooted at v, with parent edge uv* is the connected component of $T \setminus \{uv\}$ that contains v, rooted in v, equipped with the following order on the children of each node w: order the children of w in counter-clockwise order starting from the parent of w if $w \neq v$ and with the edge uv if $w = v$. We say that a and b are the *leftmost and rightmost* *elements of* $V(T_v) \cap V(\partial B)$, respectively, if $a, b \in V(T_v) \cap V(\partial B)$, $V(T_v) \cap V(\partial B) \subseteq V(\partial B[a, b])$, and the face of $\partial B \cup V(T_v)$ that contains the edge uv is incident to the edges of $\partial B[b, a]$.

3.1 Problem definitions

For completeness, we formally state the problems considered in this paper.

Planar Steiner Tree
Input: A planar graph G, a set of terminals $S \subseteq V(G)$.
Task: Find a connected subgraph T of G of minimum possible length such that $S \subseteq V(T)$ (i.e., T connects S).

Planar Steiner Forest
Input: A planar graph G, a family of pairs of terminals $\mathcal{S} \subseteq V(G) \times V(G)$.
Task: Find a subgraph H of G of minimum possible length such that for each $(s, t) \in \mathcal{S}$, the terminals s and t lie in the same connected component of H.

Observe that Planar Steiner Tree reduces to Planar Steiner Forest by taking the family $\mathcal{S}$ to be $S \times S$.

As we study Planar Edge Multiway Cut only in unweighted graphs, we state this problem in the unweighted setting only.

Planar Edge Multiway Cut (PEMwC)
Input: A planar graph G, a set of terminals $S \subseteq V(G)$.
Task: Find a minimum set of edges X such that no two terminals lie in the same connected component of $G \setminus X$.

In the bounded-genus case, we assume that the input graph is given together with an embedding into a surface of genus g such that the interior of each face is homeomorphic to an open disc.

4 The case of a nicely decomposable brick

Sections 4–9 are devoted to the proof of Theorem 1.1. However, in most places we take a more general view and argue about edge-weighted graphs, as we would like to re-use the obtained structural results in the weighted variant, discussed in Section 10. Hence, unless otherwise stated, all graphs are equipped with a weight function w.

We first give formal definitions of the brick decomposition and related notions, and proceed to define what it means for a brick to be nicely decomposable. Then we explain how Theorem 1.1 can be applied recursively.

Definition 4.1. *We say that a brick B' is* a subbrick *of a brick B if B' is a subgraph of B consisting of all edges enclosed by $\partial B'$.*

Definition 4.2. *For a brick B, a* brick covering *of B is a family $\mathcal{B} = \{B_1, B_2, \ldots, B_p\}$ of bricks, such that* (i) *each B_i, $1 \leq i \leq p$, is a subbrick of B, and* (ii) *each face of B is contained in at least one brick B_i, $1 \leq i \leq p$. A brick covering is called a* brick partition *if each face of B is contained in* exactly one *brick B_i.*

Let us now discuss the notion of brick partition. If $\mathcal{B} = \{B_1, B_2, \ldots, B_p\}$ is a brick partition of B, then it follows that every edge of ∂B belongs to the perimeter of exactly one brick B_i, while every edge of int$B$ either is in the interior of exactly one brick B_i, or lies on perimeters of exactly two bricks B_i, B_j for $i \neq j$.

Any connected set $F \subseteq B$ will be called a *connector*. Let S be the set of vertices of ∂B adjacent to at least one edge of F; the elements of the set S will be called the *anchors* of the connector F. We then say that F *connects* S. For a connector F, we say that F is *optimal* if there is no connector F' with $w(F') < w(F)$ that connects a superset of the anchors of F. Clearly, each optimal connector F induces a tree, whose every leaf is an anchor of F. For a connector F, every part of ∂B between two consecutive anchors of F will be called an *interval of* F.

We say that a connector $F \subseteq B$ is *brickable* if the boundary of every inner face of $\partial B \cup F$ is a simple cycle, i.e., these boundaries form subbricks of B. Let $\mathcal{B}$ be the corresponding brick partition of B; observe that then $\sum_{B' \in \mathcal{B}} w(\partial B') \leq w(\partial B) + 2w(F)$. Note that a tree is brickable if and only if all its leaves lie on ∂B and, consequently, every optimal connector is brickable. We now move to the definition of one of the crucial notions that explains which partitions and coverings can be used for the recursive step.

Definition 4.3. *The* total perimeter *of a brick covering $\mathcal{B} = \{B_1, \ldots, B_p\}$ is defined as $\sum_{i=1}^{p} w(\partial B_i)$. For a constant $c > 0$, $\mathcal{B}$ is* c-short *if the total perimeter of $\mathcal{B}$ is at most $c \cdot w(\partial B)$. For a constant $\tau > 0$, $\mathcal{B}$ is* τ-nice *if $w(\partial B_i) \leq (1 - \tau) \cdot w(\partial B)$ for each $1 \leq i \leq p$.*

Similarly, a brickable connector $F \subseteq B$, with $\mathcal{B} = \{B_1, \ldots, B_p\}$ the corresponding brick partition, is c-short *if $\mathcal{B}$ is c-short, simply* short *if it is 3-short, and F is* τ-nice *if $\mathcal{B}$ is τ-nice.*

Observe that for a brickable connector $F \subseteq B$, if $w(F) \leq w(\partial B) \cdot (c-1)/2$, then F is c-short, and in particular if $w(F) \leq w(\partial B)$, then F is 3-short. Moreover, if F is a tree with leaves on ∂B and of length at most $w(\partial B)$, then F is a short brickable connector. Such a tree is called a 3-*short tree* (or just *short* instead of 3-short, for simplicity). The following theorem is needed to make our proof algorithmic.

Theorem 4.4. *Let $\tau > 0$ be a fixed constant. Given an unweighted brick B, in $\mathcal{O}(|\partial B|^8 |B|)$ time one can either correctly conclude that no short τ-nice tree exists in B or find a short τ-nice brick covering of B.*

A slightly more technical variant of Theorem 4.4, in the edge-weighted setting, is stated in Section 9. The proofs of Theorem 4.4 and its edge-weighted counterpart, given in Section 9, are a technical modification of the classical algorithm of Erickson et al. [36] that computes an optimal Steiner tree in a planar graph assuming that all the terminals lie on the boundary of the infinite face. For technical reasons, we cannot ensure that if some short τ-nice tree exists, then the output of the algorithm of Theorem 4.4 will actually be a brick partition corresponding to some short τ-nice tree. Instead, the algorithm may output a brick covering, but one that is guaranteed to be short and nice for some choice of constants. Fortunately, this property is sufficient for our needs.

Armed with Theorem 4.4 and the notion of brick partition and covering, we may now describe the recursive step in the algorithm of Theorem 1.1. Thus, in the rest of this section we work with unweighted bricks only, and $w(H) = |H|$ for any subgraph H. The following lemma is the main technical contribution of this section.

Lemma 4.5. *Let $c, \tau > 0$ be constants. Let B be an unweighted brick and let $\mathcal{B} = \{B_1, \ldots, B_p\}$ be a c-short τ-nice brick covering of B. Assume that the algorithm of Theorem 1.1 was applied recursively to bricks $B_1, \ldots, B_p$, and let $H_1, \ldots, H_p$ be the subgraphs output by this algorithm for $B_1, \ldots, B_p$, respectively, where $|H_i| \leq C \cdot |\partial B_i|^\alpha$ for some constants $C > 0$ and $\alpha \geq 1$ such that $(1 - \tau)^{\alpha - 1} \leq \frac{1}{c}$. Let $H = \bigcup_{i=1}^{p} H_i$. Then H satisfies conditions* (i)-(iii) *of Theorem 1.1, with $|H| \leq C \cdot |\partial B|^\alpha$.*

Proof. To see that H satisfies condition (i), note that every edge of ∂B is in the perimeter of some brick B_i, and that $\partial B_i \subseteq H_i$ for every $i = 1, 2, \ldots, p$. Therefore, $\partial B \subseteq H$.

To see that H satisfies condition (ii), recall that $\mathcal{B}$ is c-short and that $|\partial B_i| \leq (1-\tau) \cdot |\partial B|$ for each $i = 1, 2, \ldots, p$. Therefore, $|\partial B_i|^\alpha \leq |\partial B_i| \cdot (1-\tau)^{\alpha-1} |\partial B|^{\alpha-1}$, and

$$\begin{aligned}
|H| &\leq \sum_{i=1}^{p} |H_i| \\
&\leq C \cdot \sum_{i=1}^{p} |\partial B_i|^\alpha \\
&\leq C \cdot (1-\tau)^{\alpha-1} |\partial B|^{\alpha-1} \cdot \sum_{i=1}^{p} |\partial B_i| \\
&\leq c \cdot (1-\tau)^{\alpha-1} \cdot C \cdot |\partial B|^\alpha \\
&\leq C \cdot |\partial B|^\alpha.
\end{aligned}$$

Finally, to see that H satisfies condition (iii), let $S \subseteq V(\partial B)$ be a set of terminals lying on the perimeter of B, and let T be an optimal Steiner tree connecting S in B that contains a minimum number of edges that are not in H. We claim that $T \subseteq H$. Assume the contrary, and let $e \in T \setminus H$. Since each face of B is contained in some brick of $\mathcal{B}$, there exists a brick B_i such that ∂B_i encloses e. As $\partial B_i \subseteq H_i \subseteq H$, we infer $e \notin \partial B_i$. Consider the subgraph $T \cap \mathrm{int} B_i$ (i.e., the part of T strictly enclosed by ∂B_i) and let X be the connected component of this subgraph that contains e. Clearly, X is a connector inside B_i. Since H_i is obtained by a recursive application of Theorem 1.1, there exists a connected subgraph $D \subseteq H_i$ that connects the anchors of X and that satisfies $|D| \leq |X|$. Let $T' = (T \setminus X) \cup D$. Observe that $|T'| \leq |T|$ and that T' contains strictly less edges that are not in H than T does. Since D connects the anchors of X in H_i, T' still connects the anchors of T in B, that is, T' connects S. However, T is an optimal Steiner tree that connects S, and thus T' is also an optimal Steiner tree that connects S. Since T' contains strictly less edges that are not in H than T, this contradicts the choice of T. Hence, $T \subseteq H$. □

We may now sketch the first step of our kernelization algorithm of Theorem 1.1; a formal argument is provided in Section 8. We run the algorithm of Theorem 4.4 for the brick B and some fixed small constant $\tau > 0$ (to be chosen later). If the algorithm returns a short τ-nice brick covering $\mathcal{B} = \{B_1, B_2, \ldots, B_p\}$, then we recurse on each brick B_i, obtaining a graph H_i of size bounded polynomially in $|\partial B_i|$. By Lemma 4.5, the assumptions of shortness and τ-niceness yield a polynomial bound on $|\bigcup_{i=1}^{p} H_i|$ in terms of $|\partial B|$, where the exponent α is chosen large enough so that $(1-\tau)^{\alpha-1} < \frac{1}{3}$. If the algorithm of Theorem 4.4 concluded that brick B does not contain any short τ-nice tree, then we proceed to the arguments in the next sections with this assumption.

5 Carves and the core

Let B be a possibly edge-weighted brick. We are now working with the assumption that B does not contain any short τ-nice tree for some $\tau > 0$. In this section, we define the notion of carving a small portion of the brick, which will be a crucial technical ingredient in our further reasonings. In particular, we formalize the intuition that if no short τ-nice tree can be found, then B contains a well-defined middle region, and each attempt of carving out some part of B using a limited budget cannot affect this middle region. In the following, we use a constant $\delta \in (0, \frac{1}{2})$ to be determined later.

We start by formalizing what we mean by 'carving'.

Definition 5.1. *A δ-*carve *L from a brick B is a pair (P, I), where P (called the* carvemark*) is a path in B between two distinct vertices $a, b \in V(\partial B)$ of length at most $(\frac{1}{2} - \delta) \cdot w(\partial B)$, and I (called the* carvebase*) is a shortest of the two paths $\partial B[a, b], \partial B[b, a]$. If P has only two common vertices with ∂B, i.e., $V(P) \cap V(\partial B) = \{a, b\}$, then the δ-carve is* strict*. The subgraph enclosed by the closed walk $P \cup I$ is called the* interior *of a δ-carve.*

Observe that if a δ-carve (P, I) is strict, then $P \cup I$ is a simple cycle and thus the interior of (P, I) is a brick. We often identify a strict δ-carve with this brick.

In the following lemma, we observe that if a brick does not admit any short τ-nice trees, then the carvebases cannot be much longer than the carvemarks.

Lemma 5.2. *For any $\tau, \delta \in (0, \frac{1}{2})$, $\delta > \tau$, if B admits no short τ-nice tree, then the base I of any δ-carve (P, I) in B has length at most $w(P) + \tau w(\partial B)$.*

Proof. Consider a δ-carve $L = (P, I)$ with the carvemark P between vertices a, b, such that $I = \partial B[a, b]$. Let $I' = \partial B[b, a]$. Assume on the contrary that $w(I) > w(P) + \tau w(\partial B)$. Then $w(I) \in (w(P) + \tau w(\partial B), \frac{1}{2} w(\partial B)]$. Hence, $w(I') \in [\frac{1}{2} w(\partial B), (1 - \tau) w(\partial B) - w(P))$. Clearly, P is a brickable connector in B. Let $\mathcal{B}$ be the corresponding brick partition of B. Note that each brick $B' \in \mathcal{B}$ has its perimeter contained entirely in either $P \cup I$ or $P \cup I'$. Since

$$w(P \cup I), w(P \cup I') \leq w(P) + (1 - \tau) w(\partial B) - w(P) \leq (1 - \tau) w(\partial B),$$

B' has perimeter at most $(1 - \tau) w(\partial B)$. As $w(P) \leq w(\partial B)$, P is a short τ-nice tree, a contradiction. □

By applying Lemma 5.2 to the maximum length of a carvemark, that is, $(\frac{1}{2} - \delta) \cdot w(\partial B)$, we obtain the following corollary.

Corollary 5.3. *For any $\tau \in (0, \frac{1}{4})$ and any $\delta \in [2\tau, \frac{1}{2})$, if B admits no short τ-nice tree, then the base of any δ-carve $L = (P, I)$ in B has length at most $(\frac{1}{2} - \frac{\delta}{2}) \cdot w(\partial B)$. In particular, $w(P) + w(I) \leq (1 - \frac{3}{2}\delta) w(\partial B) < (1 - \tau) w(\partial B)$.*

Note that Corollary 5.3 implies that, under its assumptions, the base of a carve is unique. Moreover, we can make the following observation. Recall that a tree T in B is brickable if and only if all its leaves lie on ∂B.

Lemma 5.4. *For any $\tau \in (0, \frac{1}{4})$ and any $\delta \in [2\tau, \frac{1}{2})$, if B admits no short τ-nice tree, then for any brickable short tree T with diameter not bigger than $(\frac{1}{2} - \delta) \cdot w(\partial B)$ there exists an interval I_T of ∂B of length at most $(\frac{1}{2} - \frac{\delta}{2}) w(\partial B)$ such that all anchors of T are in I_T.*

Proof. Observe that T is short, but not τ-nice. Hence, there exists a brick B' induced by T of perimeter bigger than $(1 - \tau) w(\partial B)$. The intersection of $\partial B'$ with T cannot be longer than the diameter of T, so $\partial B' \setminus T$, which is an interval I on ∂B, has length at least

$$(1 - \tau)\, w(\partial B) - \left(\frac{1}{2} - \delta\right) w(\partial B) = \left(\frac{1}{2} - \tau + \delta\right) w(\partial B) \geq \left(\frac{1}{2} + \frac{\delta}{2}\right) w(\partial B).$$

All other anchors of T need to be contained in the interval $I_T = \partial B \setminus I$, which is of length at most $(\frac{1}{2} - \frac{\delta}{2}) w(\partial B)$. □

We now proceed to defining the region that can be carved out by some δ-carve.

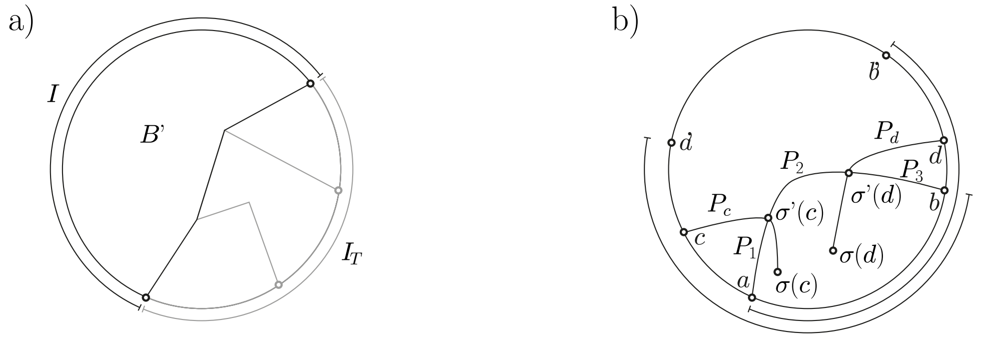


Figure 6: Panel (a) illustrates the proof of Lemma 5.4, whereas panel (b) refers to Claim 5.8.

Definition 5.5. *A subgraph F of B can be δ-*carved *if there is a δ-carve L of B such that F is also a subgraph of the interior of L.*

In particular, a vertex, edge, or face of B can be δ-carved if there is a δ-carve L of B that encloses this vertex, edge, or face. One can also define a similar notion for strict δ-carves. The following lemma shows that in the case that is of our interest, the two notions coincide.

Lemma 5.6. *For any $\tau \in (0, \frac{1}{4})$ and any $\delta \in [2\tau, \frac{1}{2})$, if a brick B admits no short τ-nice tree, then a face f of B can be δ-carved if and only if it can be strictly δ-carved.*

Proof. By definition, a face that can be strictly δ-carved can also be δ-carved. Therefore, we proceed to prove the converse. Let f be a face enclosed by a δ-carve $L = (P, I)$, where $I = \partial B[a, b]$ for two vertices $a, b \in V(\partial B)$. We may assume that $|P \cap V(\partial B)|$ is minimum among all δ-carves that δ-carve f.

We prove that $|P \cap V(\partial B)| = 2$, and thus L is strict. Suppose for sake of contradiction that $|P \cap V(\partial B)| > 2$, and let c be any internal vertex of P that lies on ∂B. We consider two cases.

In the first case, suppose that $c \in V(I)$. Observe that $L_1 = (P[a, c], \partial B[a, c])$ and $L_2 = (P[c, b], \partial B[c, b])$ are both δ-carves, because $w(\partial B[a, c]), w(\partial B[c, b]) < w(\partial B[a, b]) \leq \frac{1}{2} w(\partial B)$ and also $w(P[a, c]), w(P[c, b]) \leq w(P) \leq (\frac{1}{2} - \delta) \cdot w(\partial B)$. Moreover, at least one of these δ-carves encloses f. Since the carvemarks of L_1 and L_2 contain less vertices of ∂B than L does, we contradict the choice of L.

In the second case, suppose that $c \notin V(I)$. Observe that P is a brickable tree of diameter and size at most $(\frac{1}{2} - \delta) \cdot w(\partial B)$ that connects three anchors a, b, and c. Consequently, let I_P be the interval whose existence is asserted by Lemma 5.4 for P. As $a, b \in V(I_P)$ and $w(\partial B[b, a]) > w(\partial B)/2$ by Corollary 5.3, we have that $\partial B[a, b] \subseteq I_P$. Therefore, either $\partial B[a, c]$ or $\partial B[c, b]$ is contained in I_P, and thus has length at most $(\frac{1}{2} - \frac{\delta}{2}) w(\partial B)$. Without loss of generality, assume that it is $\partial B[a, c]$. In this case, $L' = (P[a, c], \partial B[a, c])$ is a δ-carve that encloses f, because it encloses a superset of the faces enclosed by (P, I). Since the carvemark of L' contains less vertices of ∂B than L does, we contradict the choice of L. □

We now present the main result of this section: there is a middle region of B that cannot be carved out of B using a limited budget, i.e. by a δ-carve for some appropriate choice of δ.

Theorem 5.7 (Core Theorem)**.** *For any $\tau \in (0, \frac{1}{4})$ and any $\delta \in [2\tau, \frac{1}{2})$, if B has no short τ-nice tree, then there exists a face of B that cannot be δ-carved. Moreover, such a face can be found in $O(|B|)$ time.*

Proof. We first prove the existential statement, and then show how the proof can be made algorithmic.

Define maps $v \to \pi(v)$ and $v \to \xi(v)$ for $v \in V(B)$, such that $\pi(v)$ is a vertex of ∂B closest to v, and $\xi(v)$ is a shortest path between v and $\pi(v)$. We can assume for any two vertices $v_1, v_2 \in V(B)$ that $\xi(v_1)$ and $\xi(v_2)$, when traversed from v_1 and v_2 respectively, are either disjoint, or when they meet they continue together towards the same vertex of ∂B (implying that $\pi(v_1) = \pi(v_2)$). Such a property can be ensured by constructing maps π, ξ in the following manner: attach a super-terminal s_0 adjacent to every vertex of ∂B with unit-weight edges, and apply a linear-time shortest-path algorithm [45] from s_0. In the obtained shortest-path tree, vertices of ∂B are children of the root s_0. For each subtree $T_{v'}$ rooted in a child v' of s_0, we set $\pi(v) = v'$ for every vertex $v \in T_{v'}$, and we set $\xi(v)$ as the path from v to v' in $T_{v'}$. Note that by the definition of maps π, ξ, for any $v \in V(\partial B)$, $\pi(v)$ is equal to v and $\xi(v)$ is a path of length zero that consists of the single vertex v.

Now fix some strict δ-carve $L = (P, \partial B[a, b])$, where $a, b \in V(\partial B)$ are the endpoints of the carvemark P of L. Let B' be the subbrick enclosed by L.

Claim 5.8. *There is an interval I_L on ∂B of length at most $(\frac{1}{2} - \frac{\delta}{2}) \cdot w(\partial B)$ that* (i) *contains the carvebase of L, and* (ii) *contains $\pi(v)$ for any $v \in V(B')$.*

Proof. Let $D := \pi(V(B')) \setminus V(\partial B[a, b])$. If $D = \emptyset$, then $I_L := \partial B[a, b]$ satisfies the desired conditions by Corollary 5.3, so assume otherwise. Let $\sigma : D \to V(B')$ be any mapping such that $\pi(\sigma(c)) = c$ for any $c \in D$. Note that $\xi(\sigma(c))$ intersects P; let $\sigma'(c)$ be the vertex of $V(\xi(\sigma(c))) \cap V(P)$ that is closest to c on $\xi(\sigma(c))$ and let $P_c := \xi(\sigma(c))[c, \sigma'(c)]$. Observe that, by the construction of the paths $\xi(\cdot)$, for distinct $c, d \in D$, the paths $\xi(\sigma(c))$ and $\xi(\sigma(d))$ are vertex-disjoint.

We now show that, for any $c, d \in D$ (where possibly $c = d$), there exists an interval $I_{c,d} \subseteq \partial B$ such that $w(I_{c,d}) \leq (\frac{1}{2} - \frac{\delta}{2})w(\partial B)$, $c, d \in V(I_{c,d})$ and $\partial B[a, b] \subseteq I_{c,d}$. Consider the subgraph $T_{c,d} := P \cup P_c \cup P_d$ (see Figure 6b). Observe that $T_{c,d}$ is a brickable tree in B with anchors a, b, c, and d. Without loss of generality, assume that a, $\sigma'(c)$, $\sigma'(d)$, and b lie on P in this order (possibly $\sigma'(c) = \sigma'(d)$ if $c = d$). Denote $P_1 = P[a, \sigma'(c)]$, $P_2 = P[\sigma'(c), \sigma'(d)]$, and $P_3 = P[\sigma'(d), b]$. As $\xi(\sigma(c))$ is a shortest path between $\sigma(c)$ and $V(\partial B)$, $w(P_c) \leq w(P_1)$ and, symmetrically, $w(P_d) \leq w(P_3)$. Consequently, the diameter of $T_{c,d}$ is bounded by $w(P)$, which is at most $(\frac{1}{2} - \delta)w(\partial B)$ by definition, and thus

$$w(T_{c,d}) \leq w(P) + w(P_c) + w(P_d) \leq w(P) + w(P_1) + w(P_3) \leq 2w(P) < w(\partial B).$$

Hence, Lemma 5.4 applies to $T_{c,d}$, and we obtain an interval of length at most $(\frac{1}{2} - \frac{\delta}{2})w(\partial B)$ that contains a, b, c, and d. For any $c, d \in D$, let us denote the interval obtained this way by $I_{c,d}$. As $w(\partial B[b, a]) > w(\partial B)/2$, we have $\partial B[a, b] \subseteq I_{c,d}$. Hence, $I_{c,d}$ has the claimed properties.

We now find the interval I_L. Traverse ∂B in the counter-clockwise direction from a and let b' be the last vertex for which $w(\partial B[a, b']) \leq (\frac{1}{2} - \frac{\delta}{2})w(\partial B)$. Symmetrically, traverse ∂B in the clockwise direction from b and let a' be the last vertex for which $w(\partial B[a', b]) \leq (\frac{1}{2} - \frac{\delta}{2})w(\partial B)$. Observe that a', a, b, b' lie on ∂B in this counter-clockwise direction and $a' \neq b'$. Moreover, note that for any $c, d \in D$, it follows from the properties of $I_{c,d}$ that $I_{c,d} \subseteq \partial B[a', b']$, and thus $D \subseteq \partial B[a', b']$. Let c_0 and d_0 be the vertices of D that are closest to a' and b' on $\partial B[a', b']$, respectively (possibly $c_0 = d_0$ if $|D| = 1$). We claim that $I_L := I_{c_0,d_0}$ satisfies the conditions of the claim. By the properties of I_{c_0,d_0} proven above, the length of I_L is at most $(\frac{1}{2} - \frac{\delta}{2})w(\partial B)$ and $\partial B[a, b] \subseteq I_{c_0,d_0}$. Hence, property (i) is satisfied. If $c_0 = d_0$, then $|D| = 1$, and property (ii) is satisfied by the construction of I_{c_0,d_0}. If $c_0 \neq d_0$, then $\partial B[d_0, c_0] \not\subseteq \partial B[a', b']$ and, consequently, $\partial B[c_0, d_0] \subseteq I_{c_0,d_0}$. Since $\partial B[a, b] \subseteq I_{c_0,d_0}$ and $D \subseteq \partial B[c_0, d_0]$, we infer that $\pi(V(B')) \subseteq V(I_{c_0,d_0})$. Hence, property (ii) is satisfied. This finishes the proof of the claim. ⌟

Armed with Claim 5.8, we can proceed to the proof of the existential statement of Theorem 5.7. The proof strategy is as follows: given the map $\pi : V(B) \to V(\partial B)$, we extend π to a map $\tilde{\pi}$ such that:

(i) $\tilde{\pi}$ is a continuous map from the closed disk enclosed by ∂B to its boundary;

(ii) $\tilde{\pi}$ is the identity when restricted to the boundary of this disk, i.e., to ∂B.

We will define the extension $\tilde{\pi}$ using Claim 5.8 and the assumption that every face of B can be δ-carved. Such a mapping $\tilde{\pi}$, however, would be a retraction of a closed disc onto its boundary. This contradicts Borsuk's non-retraction theorem [13], which states that such a retraction cannot exist.

We proceed with the construction of $\tilde{\pi}$. We first extend map π to the edges of B. Consider any edge vw of B. Since vw lies on the perimeter of some face of B, there exists a δ-carve L that encloses vw. By Lemma 5.6, we can assume that L is strict. By Claim 5.8, $\pi(v)$ and $\pi(w)$ both lie on I_L, which is of length at most $(\frac{1}{2} - \frac{\delta}{2}) \cdot w(\partial B)$. Hence, among the two intervals $\partial B[\pi(v), \pi(w)]$ and $\partial B[\pi(w), \pi(v)]$, one is of length at most $(\frac{1}{2} - \frac{\delta}{2}) \cdot w(\partial B)$ and one is of length at least $(\frac{1}{2} + \frac{\delta}{2}) \cdot w(\partial B)$. Therefore, we map the edge vw in a continuous manner onto the shorter of these two intervals in such a way that the distance between any two points on the embedding of vw is proportional to the distance of their images on this shorter interval. Note that the image of vw is a subinterval of I_L for every L that strictly δ-carves vw. By Claim 5.8, I_L and $I_{L'}$ for strict δ-carves L and L' can share only a subinterval. Moreover, observe that if $vw \in \partial B$, then $\pi(v) = v$, $\pi(w) = w$ and $\tilde{\pi}$ is the identity on vw. Hence, property *(ii)* of $\tilde{\pi}$ is already satisfied.

It remains to define $\tilde{\pi}$ on faces of B. Let f be any face of B. Since we assumed that every face of B can be δ-carved, there exists some δ-carve L that encloses f. Again, by Lemma 5.6, we can assume that L is strict. As we have observed, $\pi(u) \in I_L$ for every u on the boundary of f and $\tilde{\pi}(e) \subseteq I_L$ for every edge e on the boundary of f. Since I_L is an interval, which is a simply connected metric space, we can extend $\tilde{\pi}$ from the boundary of face f to its interior in a continuous manner such that the whole face f is mapped into I_L.

By construction, $\tilde{\pi}$ is continuous and maps the closed disc enclosed by ∂B onto its boundary such that ∂B is fixed in this mapping. Hence, $\tilde{\pi}$ is a retraction of a disc onto its boundary, contradicting Borsuk's non-retraction theorem. Hence, there must be an inner face of B that cannot be δ-carved and the existential statement is proved.

Finally, we present how to find such a face in time $O(|B|)$. As discussed earlier, we construct the mapping π by first placing a super-terminal s_0 on the outer face of B, attaching it to each vertex of $V(\partial B)$ with a unit-weight edge, and then constructing a shortest-path tree from s_0 in the obtained plane graph in linear time [45]. Observe now that we have in fact proven not only that some face f_0 cannot be δ-carved, but also that for some face f_0, the images of the vertices of f_0 are not contained in an interval of length at most $(\frac{1}{2} - \frac{\delta}{2}) \cdot w(\partial B)$ on ∂B — otherwise, the extended mapping $\tilde{\pi}$ could be constructed, leading to a contradiction. Clearly, given the mapping π we can identify such a face f_0 in $O(|B|)$ time by performing a linear-time check on the boundary of each face of B. By Claim 5.8, any face for which this check fails cannot be δ-carved. □

6 Mountains

In this section, we start to develop the tools that we need to find a cycle C of length $\mathcal{O}(w(\partial B))$ that lies close to the perimeter of B and that separates the core from all vertices of degree at least three of some optimal solution for any set of terminals on ∂B. To this end, we need a deep and rigorous understanding of the brick. Then, in Section 7, we exploit this understanding to actually find the cycle C.

Before we start, we need the following notion. For a path P in a brick B connecting a and b, and a real $0 \leq \kappa \leq w(P)$, we define the *vertex at distance κ from a on P*, denoted $v(P, a, \kappa)$ as follows. If there exists $v \in V(P)$ such that $w(P[a, v]) = \kappa$, then $v(P, a, \kappa) = v$. Otherwise, we find the unique edge $xy \in P$ such that $w(P[a, x]) < \kappa < w(P[a, y])$, subdivide it by inserting a new vertex v such that $w(xy) = w(xv) + w(vy)$ and $w(P[a, x]) + w(xv) = \kappa$, and set $v(P, a, \kappa) = v$. If we speak about a vertex at distance κ from a on P in B, and an edge xy needs to be subdivided to obtain $v(P, a, \kappa)$, then we abuse notation and identify the original brick B and path P with the brick B and the path P with the edge xy subdivided. Observe that this subdivision does not change any metric properties of the brick B.

The main notion in this section are δ-carves of a special form which are defined as follows.

Definition 6.1. *For a constant $\delta \in (0, 1/2)$ and fixed $l, r \in V(\partial B)$, a* δ-mountain *of B for l, r is a δ-carve M in B such that*

1. *l and r are the endpoints of the carvemark and carvebase of M;*
2. *there exists a real κ_M, $0 \leq \kappa_M \leq w(M)$, such that if we define $v_M = v(M, l, \kappa_M)$, $P_L = M[l, v_M]$ and $P_R = M[v_M, r]$, then P_L is a shortest l–P_R path in the subgraph enclosed by M and P_R is a shortest r–P_L path in the subgraph enclosed by M.*

We denote a mountain either by M to refer to the subgraph of B enclosed by the carve, or, if we want to exhibit the choice of κ_M and the partition of the carvemark into paths P_L and P_R, we write $(P_L \wedge P_R)$. By abusing notation, we may write $M = (P_L \wedge P_R)$. We call the vertex v_M the *summit* of the mountain. We also say that a mountain M *connects* the vertices l and r.

We want to stress that mountains are discrete objects. Observe that, formally, a mountain is a carve M only, and the definition speaks about the existence of a real κ_M and a vertex v_M (that may not exist in B, if we need to subdivide some edge to obtain it). Hence, a mountain is a discrete object in B, and there are only a finite number of mountains in a fixed brick B.

Throughout this section, when we discuss a (finite) family of mountains in B and prove some structural properties of them, we will assume that the summits v_M exist in B. In particular, if we use notation $M = (P_L \wedge P_R)$, then we implicitly assume that the summit v_M is (already) present in B. In the unweighted setting, one may observe that κ_M can always be taken to be integral, and then v_M always exists in the brick B. In the edge-weighted setting, we can ensure that v_M exists by subdividing some edges. Observe that subdividing some edges of B does not change the family of mountains with fixed endpoints l and r. However, when we move to the algorithmic part — where we discuss how to find some specific mountains in a brick B — we will need to be careful not to assume that v_M is present in the brick B.

Before we move on to the properties of δ-mountains, we give an intuition why we study this notion. Assume that among the terminals S lying on the boundary of the brick, one can distinguish a small set $Y \subseteq S$ that are "close enough" to each other and considerably "far away" from $S \setminus Y$. Intuitively, an optimal Steiner tree connecting S should gather all of Y in one subtree T_v such that the leftmost and rightmost elements of Y on the interval of ∂B containing Y, denote them by l and r, correspond to the leftmost and the rightmost anchors of T_v. Consider the δ-carve induced by the path in T_v joining l and r, with carvebase $\partial B[l, r]$. Observe that if this δ-carve was not a δ-mountain with summit v, then there would exist a shorter path inside this δ-carve that could be used as a shortcut to decrease the cost of T. This is formalized in the following lemma.

Lemma 6.2. *Let B be a brick and T be an optimal Steiner tree that connects $S := V(T) \cap V(\partial B)$ in B. Let $uv \in T$ be an edge of T, where v is of degree at least 3 in T, and let T_v be subtree* *of T rooted at v with parent edge uv. Let a and b be the leftmost and rightmost elements of $V(T_v) \cap V(\partial B)$ and let $l, r \in \partial B[b, a]$ be two vertices such that $l \neq r$ and $a, b \in \partial B[l, r]$. Let $P_L = T[v, a] \cup \partial B[l, a]$ and $P_R = T[v, b] \cup \partial B[b, r]$. If $w(\partial B[l, r]) < w(\partial B)/2$, then $M := (P_L \wedge P_R)$ is a δ-mountain, connecting l and r, for any $\delta < 1/2 - (w(P_L) + w(P_R))/w(\partial B)$.*

Proof. Recall that, by the definition of the leftmost and rightmost elements of $V(T_v) \cap V(\partial B)$, we have that $V(T_v) \cap V(\partial B) \subseteq V(\partial B[a, b])$. As v is of degree at least 3 in T, it is of degree at least 2 in T_v and $P_L \cap P_R = \{v\}$. Therefore, $P_L \cup P_R$ is a path and it induces a δ-carve M with carvebase $\partial B[l, r]$, as $w(\partial B[l, r]) < w(\partial B)/2$.

Suppose that M is not a mountain. Without loss of generality, there exists a path P enclosed by M that connects l with $w \in V(P_R)$ such that $V(P_R) \cap V(P) = \{w\}$ and $w(P) < w(P_L)$. By construction, P passes through a and $P[l, a] = \partial B[l, a]$. Let D be the subgraph of M enclosed by the closed walk $P_L[a, v] \cup P[a, w] \cup P_R[v, w]$. Define $T' := (T \setminus D) \cup P_R[v, w] \cup P[a, w]$. As $P_R[v, w] \cup P_L[a, v] \subseteq D$, $w(T') < w(T)$. By the definition of a and b, $P_L \setminus P$ does not contain any vertex of ∂B. Therefore, T' is a connected subgraph of B connecting $V(T) \cap V(\partial B)$, a contradiction to the minimality of T. □

The goal of this section is essentially to prove that if we take the union of all maximal δ-mountains with fixed l and r, then the perimeter of the resulting subgraph has length bounded linearly in the length of the carvebase. This intuition is captured by the following theorem.

Theorem 6.3 (Mountain Range Theorem)**.** *Fix $\tau \in [0, 1/4)$ and $\delta \in [2\tau, 1/2)$ and assume B does not admit any τ-nice 3-short tree. Then for any fixed $l, r \in V(\partial B)$ with $w(\partial B[l, r]) < w(\partial B)/2$, there exists a closed walk $W_{l,r}$ in B of length at most $3w(\partial B[l, r])$ such that, for each face f of B, f is enclosed by $W_{l,r}$ if and only if f belongs to some δ-mountain connecting l and r. Moreover, the set of the faces enclosed by $W_{l,r}$ can be computed in $O(|B|)$ time.*

The set of the faces enclosed by $W_{l,r}$ is called the *δ-mountain range* of l, r.

Observe that in Lemma 6.2, the discussed mountain has summit v that belongs to B (i.e., we do not need to subdivide any edge). However, in the edge-weighted setting we need to allow the mountains to have summits in the middle of some edges to obtain the statement of Theorem 6.3.

The rest of this section is devoted to the proof of Theorem 6.3. Henceforth, we assume that $\tau \in [0, 1/4)$, $\delta \in [2\tau, 1/2)$, $l, r \in V(\partial B)$ are fixed. Whenever we speak about a mountain, we mean a δ-mountain connecting l and r.

6.1 Preliminary simplification steps

We start the proof of Theorem 6.3 with the following simplification step. We attach to B two paths $\mathfrak{P}$, $\mathfrak{P}'$ connecting l and r, being copies of $\partial B[l, r]$ and $\partial B[r, l]$, respectively, drawn in the outer face of B in such a manner that $\mathfrak{P} \cup \mathfrak{P}'$ is the infinite face and $\mathfrak{P} \cup \partial B[l, r]$ and $\mathfrak{P}' \cup \partial B[r, l]$ are two finite faces of the constructed graph B'. Note that B' is also a brick of perimeter $w(\partial B)$, and that all δ-mountains connecting l and r in B are also δ-mountains in B' (with carvebase $\partial B[l, r]$ replaced by $\mathfrak{P}$) with the additional property that the δ-carves of these mountains are strict. Moreover, as $w(\mathfrak{P}') > w(\partial B)/2$, any mountain that is present in B' but not in B is induced by the δ-carve $(\mathfrak{P}, \mathfrak{P})$ and any choice of the summit; note that this δ-carve is enclosed by any other δ-mountain in B', and does not influence the output graph of Theorem 6.3. Hence, by somewhat abusing the notation and denoting the modified brick B' by B again, we may assume that all δ-mountains connecting l and r are induced by strict δ-carves, possibly with the exception of the trivial δ-carve $(\partial B[l, r], \partial B[l, r])$. We silently ignore the existence of the latter in the upcoming arguments, and assume that whenever we pick a mountain, it is induced by a strict δ-carve.

Hence, for any δ-mountain $M = (P_L \wedge P_R)$, the closed walk $P_L \cup P_R \cup \partial B[l,r]$ is actually a simple cycle in B, denoted ∂M.

6.2 Maximal mountains

In this subsection, we describe two properties of mountains that will be crucial in the remainder of the proof of Theorem 6.3. The first property is the following easy consequence of the definition of a mountain.

Lemma 6.4. *Let $M = (P_L \wedge P_R)$ be a mountain and let $a, b \in V(\partial M)$ be such that $\partial M[a,b]$ is contained entirely in P_L or entirely in P_R. If there exists a path Q with endpoints in a and b that is enclosed by ∂M, then $w(Q) \geq w(\partial M[a,b])$.*

Proof. By symmetry, without loss of generality assume $\partial M[a,b]$ is a subpath of P_L. Note that $Q' := \partial M[v_M, a] \cup Q \cup \partial M[b,l]$ is a path connecting l and P_R, enclosed by ∂M. Hence, $w(Q') \geq w(P_L)$ and the lemma follows. □

We now define what it means for a mountain to be maximal. Observe that since ∂M is a simple cycle for each mountain M in B, the subgraph enclosed by ∂M is defined by the set of faces of B enclosed by ∂M. A mountain M is called *maximal* if this set of faces is inclusion-wise maximal, among the set of all δ-mountains connecting l and r. Note that in the proof of Theorem 6.3 we may actually look for the union of all faces enclosed by maximal mountains.

The second property is actually a condition under which a mountain cannot be maximal.

Lemma 6.5. *Let $M = (P_L \wedge P_R)$ be a mountain. Let $u, w \in V(\partial M)$ and let P be a path between u and w such that:*

1. *P does not contain any edge strictly enclosed by ∂M and, moreover, the closed walk $\partial M[u,w] \cup P$ encloses M;*
2. *$P \neq \partial M[w,u]$;*
3. *$w(P) \leq w(\partial M[w,u])$.*

Then M is not a maximal mountain.

Proof. First note that if P is a path satisfying the assumptions of the lemma, then there exists a subpath of P also satisfying the assumptions for which no internal vertex lies on ∂M (recall that all edge weights are positive). Hence, u, w lie on the carvemark of M. Let M^* denote the carve obtained by replacing $\partial M[w,u]$ with P in the carve M. We assume that P and $u, w \in V(\partial M)$ have been chosen such that the number of faces contained in M^* is minimum (satisfying the previous assumption that P does not contain any internal vertices on ∂M). As $\partial B[l,r] \subseteq \partial M$ and $\partial M[u,w] \cup P$ encloses M, we have that u is closer to l on $P_L \cup P_R$ than w is. Since $w(P) \leq w(\partial M[w,u])$, M^* is also a δ-carve.

We now consider two cases. First, suppose that u and w both lie on P_L or both lie on P_R; by symmetry, assume that they both lie on P_L. Partition the carvemark of M^* into P_L^* and P_R^* by taking P_L^* equal to P_L with $\partial M[w,u]$ substituted by P, and taking P_R^* equal to P_R. Note that thus $w(P_L^*) \leq w(P_L)$. We claim that M^* treated as $(P_L^* \wedge P_R^*)$ is also a δ-mountain. Together with the observation that M^* encloses a proper superset of the faces enclosed by M (since no edge of P is enclosed by M), this contradicts that M is maximal.

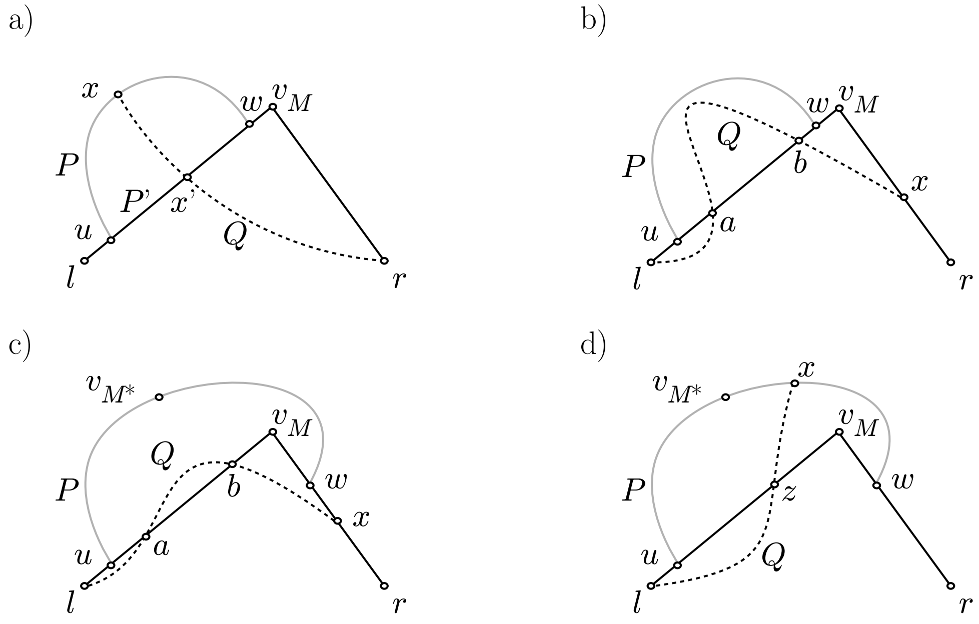


Figure 7: The cases considered in the proof of Lemma 6.5.

For sake of contradiction, assume that M^* is not a δ-mountain. Suppose that there exists a shortest path Q in M^* between r and some $x \in P^*_L$ that is shorter than $P^*_R = P_R$ — see Figure 7 (a). Observe that then Q must meet P_L, and let x' be the first point of intersection of Q and P_L, counting from r. We infer that $Q[r, x']$ is entirely contained in M and that $w(Q[r, x']) \leq w(Q) < w(P_R)$. Since $x' \in V(P_L)$, this contradicts that M is a mountain. Therefore, there exists a shortest path Q in M^* between l and some $x \in P^*_R = P_R$ that is shorter than P^*_L — see Figure 7 (b). Since $w(Q) < w(P^*_L) \leq w(P_L)$, Q must contain an edge that is not enclosed by M, since otherwise existence of Q would contradict the fact that M is a mountain. Then Q contains some subpath $Q[a, b]$ where $a, b \in V(\partial M)$ but no internal vertex of $Q[a, b]$ lies on ∂M. We can moreover assume that no edge of Q is strictly enclosed by $Q[a, b] \cup \partial M[b, a]$. As Q is a shortest path, we have that $w(Q[a, b]) \leq w(\partial M[b, a])$. By the choice of P as the path that minimizes the number of faces enclosed by M^*, we infer that Q would be a better candidate for P unless $Q[a, b] = P$ (and thus, $(a, b) = (u, w)$). By the definition of $Q[a, b]$, all edges of Q not on $Q[a, b]$ are enclosed by M. We infer that $w(P_L[l, u]) = w(Q[l, u])$, since $P_L[l, u]$ is a shortest path in M between l and u, and $Q[l, u]$ is enclosed by M. Similarly, $w(P_L[w, v_M]) = w(Q[w, x])$, since $P_L[w, v_M]$ is a shortest path between w and P_R and $Q[w, x]$ is enclosed by M. Thus we have that $w(Q) = w(Q[l, u]) + w(P) + w(Q[w, x]) = w(P_L[l, u]) + w(P) + w(P_L[w, v_M]) = w(P^*_L)$, a contradiction with the choice of Q.

Now we consider the case when u lies on P_L and w lies on P_R. As $w(\partial M^*) \leq w(\partial M)$, observe that it is possible to find a vertex v_{M^*} on P (possibly by subdividing some edge of P) such that $w(\partial M^*[v_{M^*}, l]) \leq w(P_L)$ and $w(\partial M^*[r, v_{M^*}]) \leq w(P_R)$.[3] Let $P^*_L = \partial M^*[v_{M^*}, l]$ and $P^*_R = \partial M^*[r, v_{M^*}]$. We again claim that M^* treated as $(P^*_L \wedge P^*_R)$ is a δ-mountain, which in the same manner brings a contradiction.

Assume that this is not the case, and without loss of generality suppose that there is a shortest path Q in M^* between l and $x \in P^*_R$ that is shorter than P^*_L. The case that there is a path between r and P^*_L shorter than P^*_R is symmetric. If Q does not contain any edge not enclosed by ∂M, then x must in fact lie on P_R and Q is also a shorter path than P_L in

[3] We remark here that this is the sole point in the argumentation that forces us to allow mountains with summits in the middle of some edge.

M between l and P_R, a contradiction. Assume now that Q contains a subpath $Q[a, b]$ where $a, b \in V(\partial M)$ but every internal vertex of $Q[a, b]$ is not enclosed by ∂M — see Figure 7 (c). We now employ a very similar reasoning as in the previous case. Again, we may assume that no edge of Q is strictly enclosed by $Q[a, b] \cup \partial M[b, a]$. Since Q is a shortest path, we have that $w(Q[a, b]) \leq w(\partial M[b, a])$. By the choice of P as the path that minimizes the number of faces enclosed by M^*, we infer that Q would be a better candidate for P unless $Q[a, b] = P$ (and hence $(a, b) = (u, w)$). By the definition of $Q[a, b]$, all edges of Q not on $Q[a, b]$ are enclosed by M. Since v_{M^*} lies on $P = Q[a, b]$, we infer that $x = v_{M^*} = w$. Moreover, again we have that $w(P_L[l, u]) = w(Q[l, u])$, since $P_L[l, u]$ is a shortest path in M between l and u, and $Q[l, u]$ is enclosed by M. Therefore, $w(Q) = w(Q[l, u]) + w(P) = w(P_L[l, u]) + w(P) = w(P^*_L)$, a contradiction with the choice of Q.

We are left with the case when x is not enclosed by ∂M and Q can be partitioned into $Q[l, z]$ and $Q[z, x]$, where $z \in V(\partial M)$, $z \neq x$, $Q[l, z]$ is enclosed by ∂M, and no edge of $Q[z, x]$ is enclosed by ∂M — see Figure 7 (d). Since $w(Q) < w(P^*_L) \leq w(P_L)$ and M is a mountain, $z \in V(P_L) \backslash \{v_M\}$. We note that $w(Q[l, z]) = w(P_L[l, z])$, since M is a mountain, and both $Q[l, z]$ and $P_L[l, z]$ are shortest paths in M. As $w(Q) < w(P^*_L)$, we have $w(Q[z, x]) + w(P_L[u, z]) < |P^*_L[u, v_{M^*}]$. Define $\overline{P} := Q[z, x] \cup P^*_R[x, w]$, and observe that

$$\begin{aligned} w(\overline{P}) - w(\partial M[w, z]) &= w(Q[z, x]) + w(P^*_R[x, w]) - w(\partial M[w, z]) \\ &< w(P^*_R[x, w]) + w(P^*_L[u, v_{M^*}]) - w(\partial M[w, z]) - w(P_L[u, z]) \\ &\leq w(P) - w(\partial M[w, u]). \end{aligned}$$

Since $w(P) \leq w(\partial M[w, u])$ by assumption, $w(\overline{P}) < w(\partial M[w, z])$. We infer that $\overline{P}$, instead of P, would define a carve with a strictly smaller number of faces than M^*, a contradiction; note here that $\overline{P} \neq P$, since then the left-hand side and the right-hand side of the inequality above would need to be equal. This contradicts the choice of Q. □

Corollary 6.6. *Let $M = (P_L \wedge P_R)$ be a maximal mountain with summit v_M. Then* $\text{dist}_B(v_M, l) = \text{dist}_B(P_R, l)$ *and* $\text{dist}_B(v_M, r) = \text{dist}_B(P_L, r)$.

Proof. We prove $\text{dist}_B(v_M, l) = \text{dist}_B(P_R, l)$; the other case is symmetric. Clearly, $\text{dist}_B(v_M, l) \geq \text{dist}_B(P_R, l)$, so it remains to prove an inequality in the other direction. Let P be a shortest path between P_R and l. We claim that P is actually enclosed by M; if this is the case then, by the definition of mountain, $w(P) \geq w(P_L) \geq \text{dist}_B(v_M, l)$ and the lemma is proven.

Assume the contrary, and let Q be a subpath of P with endpoints $u, w \in V(\partial M)$, such that all edges of Q are not enclosed by M and, moreover, the closed walk $\partial M[u, w] \cup P$ encloses M. By Lemma 6.5, $w(Q) > w(\partial M[w, u])$, a contradiction to the fact that P is a shortest path in B. □

6.3 Untangling maximal mountains

We now show a result that implies that the boundaries of two distinct maximal mountains $M^1 = (P^1_L \wedge P^1_R)$ and $M^2 = (P^2_L \wedge P^2_R)$ cannot cross each other (in a topological sense) more than twice, because then we can find a shortcut either inside one of the mountains (which contradicts Lemma 6.4) or outside one of the mountains (which contradicts Lemma 6.5). We assume that both summits of M^1 and M^2 are present in B, that is, the corresponding edges have already been subdivided if needed.

6.3.1 From mountains to curves

To build a topological understanding of how the two mountains interact, we build a representation of them as Jordan curves.

First, we duplicate each edge of B to obtain a brick B_2; the copies of the edges are drawn in parallel in the plane, without any other part of B_2 in between. Second, we project ∂M^1 and ∂M^2 onto B_2 in the following manner. For each $e \in \partial M^1$ ($e \in \partial M^2$) we choose one copy of e to belong to ∂M^1 (∂M^2) in B_2. If $e \in \partial M^1 \cap \partial M^2$, then one copy of e belongs to ∂M^1 and the second one to ∂M^2 in B_2, so that ∂M^1 and ∂M^2 are edge-disjoint in B_2. By abuse of notation, we often consider ∂M^1 and ∂M^2 both as walks in B and in B_2.

A vertex v is a *traversal vertex* if both ∂M^1 and ∂M^2 pass though v and they cross in v in the graph B_2; that is, among the four edges of $\partial M^1 \cup \partial M^2$ incident to v considered in counter-clockwise order around v, the odd-numbered edges belong to one mountain, and the even-numbered to the second mountain. In the process of choosing copies of an edge $e \in \partial M^1 \cup \partial M^2$, we minimize the number of traversal vertices of B_2 and, minimizing this number, we secondly minimize the number of traversal vertices of B_2 that are not equal to l or r. Clearly, if $e \in \partial M^1 \triangle \partial M^2$, the choice of the copy of e does not influence the set of transversal vertices, but the aforementioned minimization criterium regularizes the choice whenever $e \in \partial M^1 \cap \partial M^2$. In particular, we note the following.

Lemma 6.7. *No internal vertex of $\partial B[l, r]$ is a traversal vertex.*

Proof. Assume otherwise, let $x \in V(\partial B[l, r])$, $x \neq l, r$ be a traversal vertex. Consider the following change: for each edge $e \in \partial B[x, r]$, swap the copies of e that belong to ∂M^1 and ∂M^2. In this manner, x stops to be a traversal vertex, all internal vertices of $\partial B[x, r]$ are traversal vertices if and only if they were traversal vertices before the change, and r may become a traversal vertex. Thus, we either decrease the number of traversal vertices, or do not change it while decreasing the number of traversal vertices not equal to l and r. This contradicts the minimization criterium for the choice of ∂M^1 and ∂M^2 in B_2. □

Now, for each $v \in V(B_2) = V(B)$ we pick a small closed disc D_v in the plane, with the drawing of v at its centre, and with radius small enough so that D_v contains v and small starting segments of a drawing of each edge of B_2 incident to v. For $\alpha = 1, 2$, we associate the following closed Jordan curve γ^α with the cycle ∂M^α in B_2: we take the drawing of ∂M^α and for each $v \in V(\partial M^\alpha)$ we replace $D_v \cap \partial M^\alpha$ with the straight line segment S_v^α connecting the two points of $\partial D_v \cap \partial M^\alpha$. We note that $\partial D_v \cap \partial M^\alpha$ consists of exactly two points since ∂M^α is a simple cycle. Moreover, $S_v^\alpha \subseteq D_v$. Consequently, γ^α is a closed Jordan curve without self-intersections. The important properties of this construction are summarized in the following lemmata.

Lemma 6.8. *$\gamma^1 \cap \gamma^2$ consists of exactly one point in each disc D_v where v is a traversal vertex, and nothing more. Moreover, for each $p \in \gamma^1 \cap \gamma^2$, the curves γ^1 and γ^2 traverse each other in the following sense: there exists an open neighbourhood O_p of p in the plane such that $\gamma^\alpha \cap O_p$ splits $\gamma^{3-\alpha} \cap O_p$ into two connected sets for $\alpha = 1, 2$. In particular, $|\gamma^1 \cap \gamma^2|$ is finite and even.*

Proof. The first claim follows from the fact that ∂M^1 and ∂M^2 are edge-disjoint in B_2, so the points of $\partial D_v \cap \partial M^1$ and $\partial D_v \cap \partial M^2$ are pairwise distinct, and the segments S_v^1 and S_v^2 intersect if and only if v is a traversal vertex. For any traversal vertex v, if we take a small open disc O_p centred in $S_v^1 \cap S_v^2$ and contained in D_v, then $O_p \cap \gamma^1$ and $O_p \cap \gamma^2$ are two straight segments intersecting and the centre of O_p, which proves the second claim. □

Lemma 6.9. *$\gamma^1 \cap \gamma^2 \neq \emptyset$.*

Proof. Note that all finite faces incident to $\partial B[l, r]$ are enclosed by ∂M^1 and ∂M^2. Consequently, if $\gamma^1 \cap \gamma^2 = \emptyset$, then γ^1 encloses γ^2 or vice versa. Therefore, $M^1 \subseteq M^2$ or vice versa, which contradicts that M^1 and M^2 are two distinct maximal mountains. □

6.3.2 Regions, elementary regions, and their properties

Observe that since $\gamma^1 \cap \gamma^2 \neq \emptyset$ by Lemma 6.9, the curves γ^1 and γ^2 induce a set of Jordan regions in the plane; denote this set by $\mathcal{R}$. The goal of this section is to analyse $\mathcal{R}$.

Lemma 6.8 immediately implies the following.

Lemma 6.10. *For each region $R \in \mathcal{R}$, the border of R can be partitioned into an even number of subcurves $\gamma_1, \gamma_2, \ldots, \gamma_{2s}$ of positive length, appearing on the border in counter-clockwise order, where $\gamma_1, \gamma_3, \ldots, \gamma_{2s-1} \subseteq \gamma^1$ and $\gamma_2, \gamma_4, \ldots, \gamma_{2s} \subseteq \gamma^2$. The number s and the choice of the curves is unique up to a cyclic shift of the indices.*

Moreover, note that, since ∂M^1 and ∂M^2 are simple cycles, a face incident to ∂M^1 is enclosed by ∂M^1 (∂M^2) if and only if it lies to the left, if we walk along ∂M^1 (∂M^2) in counter-clockwise direction. By this observation, and by the construction of the curves γ^1 and γ^2, the following is immediate.

Lemma 6.11. *For each $\alpha = 1, 2$ and for each region $R \in \mathcal{R}$, the set $R \setminus \bigcup_{v \in V(B_2)} D_v$ is either completely enclosed by ∂M^α or no point of this set is strictly enclosed by ∂M^α.*

Lemmata 6.10 and 6.11 motivate the following definitions.

Definition 6.12 (elementary region). *We say that a region $R \in \mathcal{R}$ is* elementary *if its border can be partitioned into two curves γ_1, γ_2 with $\gamma_1 \subseteq \gamma^1$ and $\gamma_2 \subseteq \gamma^2$. That is, $s = 1$ in the statement of Lemma 6.10 for the region R.*

Definition 6.13. *We partition $\mathcal{R} = \mathcal{R}^{++} \cup \mathcal{R}^{+-} \cup \mathcal{R}^{-+} \cup \mathcal{R}^{--}$ as follows: $R \in \mathcal{R}$ belongs to $\mathcal{R}^{++} \cup \mathcal{R}^{+-}$ if and only if $R \setminus \bigcup_{v \in V(B_2)} D_v$ is enclosed by ∂M^1, and to $\mathcal{R}^{-+} \cup \mathcal{R}^{--}$ otherwise. Similarly, R belongs to $\mathcal{R}^{++} \cup \mathcal{R}^{-+}$ if and only if $R \setminus \bigcup_{v \in V(B_2)} D_v$ is enclosed by ∂M^2, and to $\mathcal{R}^{+-} \cup \mathcal{R}^{--}$ otherwise.*

We also define a *curve-arc*, which is a subcurve of γ^1 or γ^2 that connects two points of $\gamma^1 \cap \gamma^2$, but does not contain any point of this intersection as an interior point. The following property of curve-arcs is immediate from Lemma 6.8.

Lemma 6.14. *If γ is a curve-arc, then exactly two regions are incident to γ: one of these regions belongs to $\mathcal{R}^{++} \cup \mathcal{R}^{--}$, and the other to $\mathcal{R}^{+-} \cup \mathcal{R}^{-+}$.*

We now show that there, in fact, exist elementary regions.

Lemma 6.15. *There exist at least two elementary regions in $\mathcal{R}^{-+} \cup \mathcal{R}^{--}$.*

Proof. Consider the infinite region R_∞ in $\mathcal{R}$. The border of this region cannot be fully contained in one of the curves γ^1 and γ^2, because they intersect. Take any curve-arc $\gamma_1 \subseteq \gamma^1$ incident to R_∞ and cut open γ^1 by removing γ_1 to obtain a Jordan arc $\gamma^1_\times$. Order the intersection points of γ^1 with γ^2 along Jordan arc $\gamma^1_\times$. Consider now the set $\mathcal{C}^2$ of curve-arcs of γ^2 that are not enclosed by γ^1. For each Jordan arc $\gamma \in \mathcal{C}^2$ tie a pair parenthesis to its endpoints. We associate the opening parenthesis with the first endpoint of γ along $\gamma^1_\times$, whereas we associate the closing parenthesis with the second one. Observe that Jordan arcs in $\mathcal{C}^2$ cannot intersect, hence, when we list the parenthesis along $\gamma^1_\times$ we obtain a valid parenthesis expression E. We have to consider two cases.

First, suppose that the first and the last parenthesis in E belong to the same pair given by arc γ. We observe that the infinite region in $\mathcal{R}$ is elementary, as its boundary is formed by γ_1 and γ. To obtain the second elementary region, observe that there has to be a pair of innermost matching parenthesis in E corresponding to some arc γ'. The Jordan region enclosed by γ' and the part of γ^1 between the endpoints of γ' is the second elementary region not enclosed by γ^1.

Second, suppose that the first and the last parenthesis in E do not form a matching pair. Then E can be decomposed into the concatenation of two valid parenthesis expressions E_1 and E_2. Both of them need to contain a pair of innermost matching parenthesis, which induce two elementary regions. □

Note that the arguments of Lemma 6.15 can be modified to exhibit two elementary regions in $\mathcal{R}^{+-} \cup \mathcal{R}^{--}$.

We introduce some more notation with respect to regions. For a region $R \in \mathcal{R}$, we associate a closed walk $W_2(R)$ in B_2 that corresponds to the border of R in the obvious manner. Note that the walk $W_2(R)$ contains each edge of B_2 at most once (since ∂M^1 and ∂M^2 are edge-disjoint). It may visit a vertex $v \in V(B_2)$ more than once, but it never *traverses* itself in such a vertex: if we walk along $W_2(R)$ in counter-clockwise direction (defined by the border of R) and we enter a vertex v along an edge $e \in E(B_2)$, then we leave the vertex v with the edge of $W_2(R)$ incident to v being the first such edge in counter-clockwise order after e. We also define a walk $W(R)$ in B as the projection of the walk $W_2(R)$ onto B.

We say that a vertex v belongs to ∂R for some region $R \in \mathcal{R}$ (written $v \in \partial R$) if and only if the border of R intersects D_v; equivalently, if $W(R)$ visits v. Similarly, we say that a region R is incident to an edge $e \in E(B)$ or $e \in E(B_2)$ if and only if $W(R)$ or $W_2(R)$ contains e.

Consider now an elementary region $R \in \mathcal{R}$. According to the definition, its border splits into curves γ_1 and γ_2, where $\gamma_\alpha \subseteq \gamma^\alpha$ for $\alpha = 1, 2$. Consequently, since ∂M^1 and ∂M^2 are simple cycles in B and B_2, the walk $W_2(R)$ splits into paths $P_2^1(R)$ and $P_2^2(R)$ in B_2 and the walk $W(R)$ splits into paths $P^1(R)$ and $P^2(R)$ in B, where $P_2^\alpha(R)$ and $P^\alpha(R)$ are a subpath of ∂M^α in B_2 and B, respectively.

Using this notation, we can present the following implication of the minimization criterium assumed in the projection of ∂M^1 and ∂M^2 onto B_2.

Lemma 6.16. *For any elementary region $R \in \mathcal{R}$, there exists a face f_2 of B_2 enclosed by $W_2(R)$ that is not a face between two copies of an edge of B, and thus, there exists a face f of B enclosed by $W(R)$.*

Proof. If such faces f_2 and f do not exist, then $P^1(R) = P^2(R)$, as $P^1(R)$ and $P^2(R)$ are simple paths. Let $v_0, e_1, v_1, e_2, \ldots, e_s, v_s$ be the vertices and edges of $P^1(R) = P^2(R)$, and let $P_2^\alpha(R) = v_0, e_{1,\alpha}, v_1, e_{2,\alpha}, \ldots, e_{s,\alpha}, v_s$ for $\alpha = 1, 2$; the edges $e_{i,1}$ and $e_{i,2}$ are the two copies of e_i in B_2. As R is a region, γ^1 and γ^2 intersect in D_{v_0} and D_{v_s}, hence v_0 and v_s are traversal vertices. Consider the following modification to ∂M^1 and ∂M^2 in B_2: for each $1 \leq i \leq s$, we swap $e_{i,1}$ with $e_{i,2}$, so that $e_{i,1}$ now belongs to ∂M^2 and $e_{i,2}$ belongs to ∂M^1. After this operation, for any $1 \leq i < s$, the vertex v_i is a traversal vertex if and only if it was a traversal vertex before the operation, while v_0 and v_s discontinue to be traversal vertices. This contradicts the minimization criterium for the choice of ∂M^1 and ∂M^2 in B_2. □

6.3.3 Two maximal mountains form a range

Intuitively, elementary regions that are finite and do not belong to $\mathcal{R}^{++}$ often give grounds to applying Lemma 6.5 and to the conclusion that M^1 or M^2 is not maximal. In this argumentation, we need to watch out for the following special case.

Definition 6.17 (cushion)**.** *An elementary region $R \in \mathcal{R}$ is called a* cushion *if $W(R)$ contains two copies of $\partial B[l, r]$ (one from ∂M^1 and one from ∂M^2), and $W_2(R)$ contains all edges of $\partial M^1 \cup \partial M^2$ incident to l or r.*

We are now ready for a crucial definition that is necessary to prove Theorem 6.3.

Definition 6.18 (range)**.** *We say that M^1 and M^2* form a range *when the following condition hold*

- *there is exactly one region R^{+-} in $\mathcal{R}^{+-}$ and one region R^{-+} in $\mathcal{R}^{-+}$;*
- *R^{+-} and R^{-+} are elementary and neither of them is a cushion;*
- *$v_{M^2} \in V(W(R^{-+})) \setminus V(P^1(R^{-+}))$ and $v_{M^1} \in V(W(R^{+-})) \setminus V(P^2(R^{+-}))$.*

The main step of the proof of Theorem 6.3, which we take in this section, is to show that every pair of maximal δ-mountains forms a range. Observe that a necessary condition for M^1 and M^2 to form a range is that γ^1 and γ^2 cross only in two points. The following lemma is used to establish this condition.

Lemma 6.19. *If there exist two elementary regions $R_1, R_2 \in \mathcal{R}$ that have a common incident curve-arc, then $|\gamma^1 \cap \gamma^2| = 2$.*

Proof. Let γ be the common incident curve-arc between R_1 and R_2, and let a and b be its endpoints. Without loss of generality, we assume that $\gamma \subseteq \gamma^1$. We have that $\partial R_1 \setminus \gamma \subseteq \gamma^2$ and $\partial R_2 \setminus \gamma \subseteq \gamma^2$, so $\gamma^2 = (\partial R_1 \cup \partial R_2) \setminus \gamma$. Hence, γ^1 and γ^2 cross only in a and b. □

We can now split the possible configurations of M^1 and M^2 into following cases.

Lemma 6.20. *One of the following holds:*

(i) there exists a finite elementary region $R \in \mathcal{R}^{--}$;

(ii) there exists an elementary region $R \in \mathcal{R}^{+-} \cup \mathcal{R}^{-+}$, such that $v_{M^1} \notin V(W(R)) \setminus V(P^2(R))$ and $v_{M^2} \notin V(W(R)) \setminus V(P^1(R))$;

(iii) the infinite region $R_\infty \in \mathcal{R}$ is elementary and is not incident to v_{M^1} nor v_{M^2};

(iv) there exists a finite elementary region that is a cushion;

(v) M^1 and M^2 form a range.

Proof. From Lemma 6.15 we know that there exist two elementary regions $R^1_1, R^1_2 \in \mathcal{R}^{--} \cup \mathcal{R}^{-+}$ and two elementary regions $R^2_1, R^2_2 \in \mathcal{R}^{--} \cup \mathcal{R}^{+-}$. Regions R^1_i and R^2_j may be sometimes equal. Up to symmetry, we have following cases.

Case *$R^1_1 = R^2_1$ and $R^1_2 = R^2_2$:* In this case, both $R^1_1 = R^2_1$ and $R^1_2 = R^2_2$ belong to $\mathcal{R}^{--}$. Hence, one of these two elementary regions is not infinite and Case (i) holds.

Case *$R^1_1 = R^2_1$ and $R^1_2 \neq R^2_2$:* If $R^1_1 = R^2_1$ is not infinite, then Case (i) holds. Hence, assume the contrary, which implies that the infinite region is elementary. Now, neither R^1_2 nor R^2_2 can be infinite. On the other hand, if one of them belongs to $\mathcal{R}^{--}$ then Case (i) holds. We are left with the case $R^1_2 \in \mathcal{R}^{-+}$ and $R^2_2 \in \mathcal{R}^{+-}$. By Lemma 6.14, R^1_2 and R^2_2 are not incident to a common curve-arc. We note that, as ∂M^1 and ∂M^2 are simple cycles, only one region $R \in \{R^1_2, R^2_2\}$ may satisfy $v_{M^1} \in V(W(R)) \setminus V(P^2(R))$ and only one region $R \in \{R^1_2, R^2_2\}$ may satisfy $v_{M^2} \in V(W(R)) \setminus V(P^1(R))$. Hence, if Case (ii) does not hold for both R^1_2 and R^2_2, we need to have that $v_{M^1} \in V(W(R^1_2)) \setminus V(P^2(R^1_2))$ and $v_{M^2} \in V(W(R^2_2)) \setminus V(P^1(R^2_2))$ or vice versa (i.e., with the roles of R^1_2 and R^2_2 swapped). In particular, neither v_{M^1} nor v_{M^2} is a traversal vertex. If Case (iii) does not hold, then since the infinite region $R^1_1 = R^2_1$ is elementary, either v_{M^1} or v_{M^2} has to be on the border of the infinite region $R^1_1 = R^2_1$. This implies that either R^1_2 or R^2_2 shares a curve-arc with $R^1_1 = R^2_1$. By applying Lemma 6.19 to these two incident elementary regions we know that γ^1 and γ^2 cross exactly twice. Consequently, each set $\mathcal{R}^{++}$, $\mathcal{R}^{+-}$, $\mathcal{R}^{-+}$ and $\mathcal{R}^{--}$ has size exactly one and all regions in $\mathcal{R}$ are elementary. Moreover, as v_{M^1} or v_{M^2} is on the border of the infinite region $R_1^1 = R_1^2$, we infer that in fact $v_{M^1} \in V(W(R_2^2)) \setminus V(P^2(R_2^2))$ and $v_{M^2} \in V(W(R_2^1)) \setminus V(P^1(R_2^1))$. If R_2^1 or R_2^2 is a cushion, we have Case (iv). Otherwise, $R^{+-} = R_2^2$ and $R^{-+} = R_2^1$ fulfills Definition 6.18.

Case *All four $R_1^1, R_1^2, R_2^1, R_2^2$ are different.:* If at least two of these regions belong to $\mathcal{R}^{--}$, then one is finite and we have Case (i). Therefore, at least three of the regions belong to $\mathcal{R}^{+-} \cup \mathcal{R}^{-+}$. Lemma 6.14 implies that at most one of them has $v_{M^1} \in V(W(R)) \setminus V(P^2(R))$ and at most one has $v_{M^2} \in V(W(R)) \setminus V(P^1(R))$. Therefore, at least one of the regions satisfies Case (ii). □

In the next lemmata we show that when one of the Cases i–iv of Lemma 6.20 holds, then either M^1 or M^2 is not maximal. Our main tools in the upcoming arguments are Lemmata 6.4 and 6.5.

Lemma 6.21. *If Case (i) in Lemma 6.20 holds, then M^1 or M^2 is not maximal.*

Proof. Let R be the elementary region R promised by Case (i). By Lemma 6.16, $W(R)$ encloses at least one finite face of B and $P^1(R) \neq P^2(R)$. If $w(P^1(R)) \leq w(P^2(R))$, then we can apply Lemma 6.5 to $P^1(R)$ and M^2, implying that there exists a mountain that strictly contains M^2. Otherwise, i.e. if $w(P^1(R)) > w(P^2(R))$, then we can apply Lemma 6.5 to $P^2(R)$ and M^1, implying that there exists a mountain that strictly contains M^1. □

Lemma 6.22. *If Case (ii) in Lemma 6.20 holds, then M^1 or M^2 is not maximal.*

Proof. Let R be the elementary region R promised by Case (ii). By Lemma 6.16, $W(R)$ encloses at least one finite face of B and $P^1(R) \neq P^2(R)$. Without loss of generality, assume that $R \in \mathcal{R}^{-+}$. If $w(P^1(R)) \geq w(P^2(R))$, then we can apply Lemma 6.5 to $P^2(R)$ and M^1, implying that there exists a mountain that strictly contains M^1. Hence, we are left with the case $w(P^1(R)) < w(P^2(R))$.

Note that $P^1(R)$ is enclosed by ∂M^2. Let $v_1, v_2, \ldots, v_s$ be the vertices of $V(P^1(R)) \cap V(P^2(R))$, in the order of their appearance on $P^2(R)$. Note that $s \geq 2$, as v_1, v_s are the endpoints of $P^1(R)$ and $P^2(R)$. Moreover, $v_1, v_2, \ldots, v_s$ is also the order of the appearance of vertices of $V(P^1(R)) \cap V(P^2(R))$ on $P^1(R)$, as $P^1(R)$ is a simple path and is enclosed by ∂M^2. As $w(P^1(R)) < w(P^2(R))$, there exists an index $1 < i \leq s$ such that $w(P^1(R)[v_{i-1}, v_i]) < w(P^2(R)[v_{i-1}, v_i])$.

By the properties of Case (ii), v_{M^2} is not in $V(P^2(R)) \setminus V(P^1(R))$; in particular, v_{M^2} is not an internal vertex of $P^2(R)[v_{i-1}, v_i]$. As $R \in \mathcal{R}^{-+}$, that is, $R \setminus \bigcup_{v \in V(B^2)} D_v$ is not enclosed by ∂M^1, and since $W(R)$ encloses $C := P^1(R)[v_{i-1}, v_i] \cup P^2(R)[v_{i-1}, v_i]$, C cannot enclose any face incident to an edge of $\partial B[l, r]$. Consequently, $P^2(R)[v_{i-1}, v_i]$ is a subpath of P_L^2 or P_R^2. However, as $w(P^1(R)[v_{i-1}, v_i]) < w(P^2(R)[v_{i-1}, v_i])$ and $P^1(R)[v_{i-1}, v_i]$ is enclosed by ∂M^2, this contradicts Lemma 6.4 and finishes the proof of the lemma. □

Lemma 6.23. *If Case (iii) in Lemma 6.20 holds, then M^1 or M^2 is not maximal.*

Proof. Let $R = R_\infty$ be the elementary infinite region promised by Case (iii). Let a and b be the endpoints of $P^1(R)$ and $P^2(R)$, and let $Q^\alpha = \partial M^\alpha \setminus P^\alpha(R)$ for $\alpha = 1, 2$. Note that $Q^1 \neq P^2(R)$ (and, symmetrically, $Q^2 \neq P^1(R)$), as otherwise ∂M^1 encloses M^2; however, in this case γ^1 and γ^2 would be disjoint, due to the minimization criterium used in the construction of ∂M^1 and ∂M^2 in B_2. Consequently, if $w(Q^1) \geq w(P^2(R))$ or $w(Q^2) \geq w(P^1(R))$, then we may apply Lemma 6.5 either to the pair $(P^2(R), M^1)$ or to the pair $(P^1(R), M^2)$, finishing the proof of the lemma. Hence, we are left with the case $w(Q^1) < w(P^2(R))$ and $w(Q^2) < w(P^1(R))$.

By Lemma 6.7, for exactly one $\alpha \in \{1,2\}$ all edges of the path $\partial M^\alpha[l,r]$ are incident to the infinite face in B_2, and all edges of $\partial M^{3-\alpha}[l,r]$ are not incident to the infinite face in B_2. Moreover, neither a nor b is an internal vertex of $\partial B[l,r]$. Consequently, at least one of the paths $P^1(R)$ and $P^2(R)$ does not contain any edge of $\partial B[l,r]$. Without loss of generality, assume it is $P^1(R)$. Moreover, by the properties of Case (iii), $P^1(R)$ does not contain v_{M^1}. Hence, $P^1(R)$ is a subpath of P^1_L or P^2_R. However, $w(Q^2) < w(P^1(R))$ and Q^2 is enclosed by ∂M^1. This contradicts Lemma 6.4. □

Lemma 6.24. *If Case (iv) in Lemma 6.20 holds, then M^1 or M^2 is not maximal.*

Proof. Let R be the cushion promised by Case (iv). By the definition of a cushion, $\partial B[l,r]$ is a subpath of both $P^1(R)$ and $P^2(R)$. As $W_2(R)$ encloses all faces of B_2 between the copies of the edges of $\partial B[l,r]$, $R \in \mathcal{R}^{+-} \cup \mathcal{R}^{-+}$. Without loss of generality assume that $P^1_2(R)[l,r]$ is incident to the infinite face of B_2 and thus $R \in \mathcal{R}^{+-}$. Let a be the endpoint of $P^1(R)$ that lies closer to l than to r, and let b be the other endpoint; note that also on $P^2(R)$ the endpoint a is closer to l than to r.

Assume that $P^1(R)[a,l] = P^2(R)[a,l]$. Consider the following operation: for each edge e of $P^1(R)[a,l]$, we swap which copy of e in B_2 belongs to ∂M^1 and which to ∂M^2. In this manner, an internal vertex of $P^1(R)[a,l]$ is a traversal vertex if and only if it was traversal vertex prior to the operation, whereas a discontinues to be a traversal vertex and l becomes a traversal vertex. Consequently, the operation does not change the total number of traversal vertices while strictly decreasing the number of traversal vertices that are not equal to l or r, a contradiction to the choice of ∂M^1 and ∂M^2 in B_2.

We infer that the closed walks $P^1(R)[a,l] \cup P^2(R)[a,l]$ and $P^1(R)[r,b] \cup P^2(R)[r,b]$ enclose each at least one face of B. The vertex v_{M^1} cannot lie both on $P^1(R)[a,l]$ and $P^1(R)[r,b]$; without loss of generality assume it does not lie on $P^1(R)[a,l]$, and $P^1(R)[a,l]$ is a subpath of P^1_L. If $w(P^1(R)[a,l]) \leq w(P^2(R)[a,l])$ then we may apply Lemma 6.5 to the pair $(P^1(R)[a,l], M^2)$. Otherwise, $w(P^2(R)[a,l]) < w(P^1(R)[a,l])$. However, $P^2(R)[a,l]$ is enclosed by ∂M^1 and $P^1(R)[a,l]$ is a subpath of P^1_L. This contradicts Lemma 6.4. □

As a consequence of the above lemmata, we infer the following.

Corollary 6.25. *Any two distinct maximal mountains form a range.*

6.4 The range of all maximal mountains

We now analyse the structure of all maximal mountains, using the crucial property established in Corollary 6.25 that any two distinct maximal mountains form a range. Recall that, formally, a mountain is only a carve in B, and therefore, there is only a finite number of mountains. Hence, we may assume that some edges of B have been subdivided, so that each mountain with endpoints l and r can choose its summit among the vertices of B.

We start with the following observation that the mountain range relation implies an order on the set of maximal mountains.

Lemma 6.26. *Let two mountains $M^1 = (P^1_L \wedge P^1_R)$ and $M^2 = (P^2_L \wedge P^2_R)$ form a range. Then $w(P^1_L) < w(P^2_L)$ or $w(P^2_L) < w(P^1_L)$. Moreover, if $w(P^1_L) < w(P^2_L)$, then $P^1(R^{-+})$ is a subpath of $\partial B[l,r] \cup P^1_R$, where R^{-+} is the elementary region in $\mathcal{R}^{-+}$.*

Proof. Consider the unique regions $R^{-+} \in \mathcal{R}^{-+}$ and $R^{+-} \in \mathcal{R}^{+-}$. Note that, by Lemma 6.14, they do not share any curve-arc that makes up their borders and, consequently, for $\alpha = 1,2$, $P^\alpha(R^{+-})$ and $P^\alpha(R^{-+})$ are edge-disjoint. Moreover, by definition of forming a range, v_{M^2} does not lie on $P^2(R^{+-})$ and v_{M^1} does not lie on $P^1(R^{-+})$.

We also infer from Lemma 6.14 that, since $|\mathcal{R}^{-+}| = |\mathcal{R}^{+-}| = 1$, there are only four curve-arcs, all incident to R^{-+} or R^{+-} and, consequently, $\mathcal{R}^{--}$ consist only of the infinite region R_∞ and $\mathcal{R}^{++}$ consists only of one region R^{++}.

Consider the faces of B_2 between the copies of the edges of $\partial B[l, r]$. By Lemma 6.7, they are all enclosed by $W_2(R)$ for a single region R. Moreover, as they are enclosed by only one of ∂M^1 and ∂M^2 in B_2, $R = R^{+-}$ or $R = R^{-+}$. As neither R^{+-} nor R^{-+} is a cushion, exactly one of the vertices l and r is a traversal vertex, and an endpoint of all four paths $P^1(R^{+-})$, $P^2(R^{+-})$, $P^1(R^{-+})$ and $P^2(R^{-+})$. We note that we may assume r to be the traversal vertex, as the other case can be reduced to this one by swapping the copies of the edges of $\partial B[l, r]$ in B_2 between ∂M^1 and ∂M^2. Moreover, by symmetry between M^1 and M^2, without loss of generality we may assume that the faces of B_2 between the copies of the edges of $\partial B[l, r]$ are enclosed by $W_2(R^{+-})$; this implies that $\partial M^1[l, r]$ is incident to the infinite face of B_2. Hence, l lies on $P^2(R^{+-})$, that is, $P^1(R^{+-})[l, r] = P^2(R^{+-})[l, r] = \partial B[l, r]$. Let a be the intersection of γ^1 and γ^2 different than r, and, at the same time, the endpoint of the paths $P^1(R^{+-})$, $P^2(R^{+-})$, $P^1(R^{-+})$ and $P^2(R^{-+})$. As v_{M^1} lies on $P^1(R^{+-})$, the path $P^2(R^{+-})[l, a]$ is a path connecting l with P^1_R that is enclosed by ∂M^1. Consequently, $w(P^1_L) \leq w(P^2(R^{+-})[l, a]) < w(P^2_L)$, where the last inequality follows from the fact that v_{M^2} lies on $P^2(R^{-+})$ and $v_{M^2} \neq a$, thus $P^2(R^{+-})[l, a]$ is a proper subpath of P^2_L. The second part of the lemma is immediate from the above discussion.

Note that if we would assume that the faces of B_2 between the copies of the edges of $\partial B[l, r]$ are enclosed by $W_2(R^{-+})$, the roles of M^1 and M^2 would change in the above reasoning and we would obtain $w(P^2_L) < w(P^1_L)$. This concludes the proof of the lemma. □

Now we proceed to analyse the union of all maximal mountains.

Lemma 6.27. *There exists a closed walk W of length at most $3w(\partial B[l, r])$ such that W encloses a face f if and only if f is contained in some maximal mountain.*

Proof. Let $\{M^i = (P^i_L, P^i_R)\}_{i=1}^s$ be the set of all maximal mountains such that $w(P^i_L) < w(P^j_L)$ for $1 \leq i < j \leq s$. By induction, we show closed walks $W^1, W^2, \ldots, W^s$ such that for each $i = 1, 2, \ldots, s$, the following holds:

1. W^i contains $P^i_R \cup \partial B[l, r]$ as a subpath.
2. If we define $\widehat{\gamma}^i$ to be the closed curve in the plane Π obtained by traversing W^i in the direction so that the $P^i_R \cup \partial B[l, r]$ is traversed from l to v_{M^i}, then, for any face f of B and any point c in the interior of f:
   - if f belongs to one of the mountains $M^1, M^2, \ldots, M^i$ then $\widehat{\gamma}^i$ is a positive element of the fundamental group $\Gamma_c \cong \mathbb{Z}$ of $\Pi \setminus \{c\}$;
   - otherwise, $\widehat{\gamma}^i$ is the neutral element of this group.

   In particular, W^i encloses f if and only if f is contained in one of the mountains $M^1, M^2, \ldots, M^i$.
3. $w(W^i) \leq w(P^1_R) + w(P^i_L) + w(\partial B[l, r])$.

Here, property 2 formalizes the intuition that maximal mountains look as they do in Figure 8. In reality, the boundaries of the mountains may actually intersect often (but not cross more than twice), which is why we need this formal property.

For $i = 1$, the induction hypothesis holds by taking $W^1 := \partial M^1$. Now assume that the induction hypothesis holds for W^i. Consider mountains M^i and M^{i+1} and apply Lemma 6.26 to them; by abuse of notation, we denote the appropriate paths as P^i and P^{i+1} instead of P^1 and P^2.

From Corollary 6.25 and Definition 6.18, we know that there exists a unique region $R \in \mathcal{R}^{-+}$. Recall that $P_R^i \cup \partial B[l,r]$ is a subpath of W^i, and that $w(P_L^i) < w(P_L^{i+1})$ by the chosen order. Hence, by Lemma 6.26, $P^i(R)$ is a subpath of W^i. We define W^{i+1} as W^i with $P^i(R)$ replaced with $P^{i+1}(R)$. Moreover, as $v_{M^{i+1}}$ lies on $P^{i+1}(R)$, it follows that $P_R^{i+1} \cup \partial B[l,r]$ is a subpath of W^{i+1}.

Let γ_W^{i+1} be the closed curve obtained by traversing $W(R)$ in counter-clockwise direction, that is in γ_W^{i+1} the path $P^i(R)$ is traversed from the endpoint closer to v_{M^i} to the endpoint closer to or on the carvebase $\partial B[l,r]$. Consider any face f of B and any point c in its interior. Note that, in the fundamental group $\Gamma_c \cong \mathbb{Z}$ of $\Pi \backslash \{c\}$, we have $\widehat{\gamma}^i + \gamma_W^{i+1} = \widehat{\gamma}^{i+1}$. If f is enclosed by γ_W^{i+1}, that is, by $W(R)$, then γ_W^{i+1} is a positive element of Γ_c, and otherwise it is the neutral element. We infer that the second condition is satisfied for the curve $\widehat{\gamma}^{i+1}$, due to the induction hypothesis, and since $W(R)$ encloses a face f if and only if f is contained in M^{i+1}, but not in M^i.

Thus, to finish the proof of the induction step we need to show the bound on the length of W^{i+1}.

Define $b = w(P_R^{i+1})$ and $e = w(P_L^i)$. Let v be the first point on P_L^{i+1} that lies on P_R^i. We denote the distance (along P_L^{i+1}) from l to v as d and the distance from v to $v_{M^{i+1}}$ as a. Finally, we denote by c the distance (along P_R^i) from r to v. These definitions are illustrated in Figure 8.

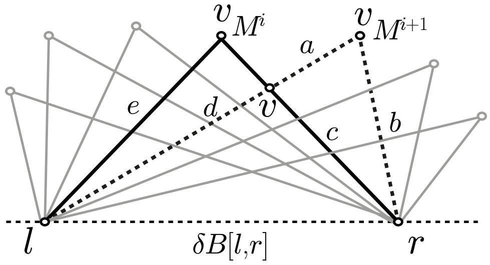


Figure 8: (Figure 4a repeated) Illustration of the inductive proof in Lemma 6.27.

Observe that $d \geq e$ because M^i is a mountain. Similarly, observe that $c \geq b$ because M^{i+1} is a mountain. Hence, we have

$$w(W^{i+1}) - w(W^i) = a + b - c \leq a \leq a + d - e = w(P_L^{i+1}) - w(P_L^i).$$

Using the induction hypothesis with the above inequality we obtain

$$\begin{aligned} w(W^{i+1}) &= w(W^{i+1}) - w(W^i) + w(W^i) \\ &\leq w(P_R^1) + w(P_L^i) + w(\partial B[l,r]) + w(P_L^{i+1}) - w(P_L^i) \\ &= w(P_R^1) + w(P_L^{i+1}) + w(\partial B[l,r]). \end{aligned}$$

This proves the induction. Hence, $W := W^s$ satisfies the conditions of the lemma as

$$\begin{aligned} w(W^s) &\leq w(P_R^1) + w(P_L^s) + w(\partial B[l,r]) \\ &\leq w(\partial B[l,r]) + w(\partial B[l,r]) + w(\partial B[l,r]) \\ &= 3w(\partial B[l,r]). \end{aligned}$$

□

We remark here that, although formally the reasoning of Lemma 6.27 has been done in the presence of all summits of mountains, the obtained walk W projects back to the original brick B, where edges have not been subdivided.

6.5 Finding the mountain range

Finally, we show the algorithm to compute the δ-mountain range. We need the following technical observation. Consider a plane drawing of B in which the segment $\partial B[l,r]$ is drawn as a horizontal segment and l is the left end of it. The *leftmost shortest* path from l to v is the shortest path that lies as much as possible to the left in the drawing of B. Symmetrically, we define the *rightmost shortest* path from r to v. Note that these notions are well defined, as they correspond to taking furthest counter-clockwise and clockwise objects around l and r, respectively, in the semi-plane above segment $\partial B[l,r]$ that contains brick B.

Observe the following connection between left- and rightmost shortest paths and maximal mountains.

Lemma 6.28. *For fixed $xy \in B$, there exists at most one maximal δ-mountain $M^{x,y}$ of B that can choose the summit on the edge xy (possibly in x or y) and has x closer to l on the carvemark $M^{x,y}$ than y. Moreover, the carvemark of $M^{x,y}$ consists of the leftmost shortest path from l to x in B, the edge xy and the rightmost shortest path from r to y in B.*

Proof. Let M be a maximal mountain that contains xy on the carvemark M, such that there exists a witness κ_M with $w(M[l,x]) \leq \kappa_M \leq w(M[l,y])$. Let P be any shortest path between l and x in B. We claim that P is enclosed by M, and a symmetrical claim holds for any shortest path between r and y in B. Note that this statement would conclude the proof of the lemma.

Assume the contrary, and let $Q = P[a,b]$ be any subpath of P whose all edges and internal vertices are not enclosed by M, but both endpoints a, b of Q lie on the carvemark M. By Lemma 6.5, $w(Q) > w(M[a,b])$ or $w(Q) > w(M[b,a])$, a contradiction to the assumption that P is a shortest path. The arguments for paths connecting y and r are symmetric. □

Observe that for a fixed vertex $u \in V(\partial B)$, the union of all leftmost shortest paths from u to every $v \in V(B)$ is a shortest-path tree rooted at u; we call it the *leftmost shortest-path tree* rooted at u. An analogous claim holds for rightmost shortest paths by symmetry. We show that this shortest-path tree can be found efficiently for fixed u.

Lemma 6.29. *For any fixed $u \in V(\partial B)$, the leftmost shortest-path tree rooted at u (and, symmetrically, the rightmost one) can be found in $O(|B|)$ time.*

Proof. The approach is the same as one proposed by Klein [50]. First, we find a shortest path tree from u in linear time [45]. Let $d(v)$ denote the distance from u to v for any $v \in V(B)$. Let H be the following directed graph. The vertex set of H is $V(B)$. Then H contains the arc (v,w) if and only if vw is an edge of B and $d(w) = d(v) + w(wv)$. Observe that H is acyclic, as all edge weights are positive. Now it suffices to find a leftmost search tree (see e.g. [64]) in H. This can be done in linear time using a simple depth-first search, which visits the neighbours of a vertex in left-to-right order. By the construction of H, this immediately translates into a leftmost shortest path tree in B. A rightmost shortest-path tree can be found symmetrically. □

In the next lemma, we make use of the left- and rightmost shortest-path trees and conclude the proof of Theorem 6.3.

Lemma 6.30. *The union of all finite faces of δ-mountains for fixed l, r can be computed in $O(|B|)$ time.*

Proof. Using Lemma 6.29 we compute the leftmost shortest-path tree rooted at l and the rightmost shortest-path tree rooted at r. Denote these trees T_l and T_r, respectively.

By traversing the tree T_l from the root to its leaves, we compute for each $v \in V(B)$ the value

$$d_l(v) = \min\{\mathrm{dist}_B(v', r) : v' \text{ is an ancestor of } v \text{ in the tree } T_l\}.$$

Symmetrically, we compute values $d_r(v)$ in the tree T_r (taking into account distances to l). This takes $O(|B|)$ time.

Let Z be the set of pairs $(x, y) \in V(B) \times V(B)$ such that $xy \in B$, $\mathrm{dist}_B(x, l) \leq d_r(y)$, $\mathrm{dist}_B(y, r) \leq d_l(x)$ and $d_r(y) + d_l(x) \geq \mathrm{dist}_B(x, l) + w(xy) + \mathrm{dist}_B(y, r)$. For every $(x, y) \in Z$, consider a walk $M^{x,y}$ in B that consists of the leftmost shortest path from l to x (i.e., the path from x to the root l in T_l), the edge xy and the rightmost shortest path from r to y (i.e., the path from y to the root r in T_r). We observe the following equivalence, captured in the next two claims.

Claim 6.31. *For every $(x, y) \in Z$, $M^{x,y}$ is a mountain.*

Proof. First observe that $M^{x,y}[l, x]$ and $M^{x,y}[y, r]$ cannot share a vertex, as otherwise $d_r(y) + d_l(x) \leq w(M^{x,y}[l, x]) + w(M^{x,y}[y, r]) = \mathrm{dist}_B(x, l) + \mathrm{dist}_B(y, r)$, a contradiction to the properties of the pairs in Z and the fact that $w(xy) > 0$. Hence, $M^{x,y}$ is a path.

We claim that $M^{x,y}$ is a mountain for $\kappa = d_r(y)$. By the properties of pairs in Z we have $w(M^{x,y}[l, x]) \leq \kappa \leq w(M^{x,y}[l, y])$ and, consequently, the candidate summit $v := v(M^{x,y}, l, \kappa)$ is located on the edge xy (possibly at one of the endpoints). If needed, subdivide the edge xy with the vertex v. As $\mathrm{dist}_B(x, l) \leq d_r(y)$, by the definition of $d_r(y)$, we have $\mathrm{dist}_B(l, M^{x,y}[v, r]) \geq d_r(y)$. Regarding the distances from r, first observe that $w(vy) + \mathrm{dist}_B(y, r) = w(M^{x,y}[v, r])$ and, hence, any path in B connecting r and v that passes through y is of length at least $w(M^{x,y}[v, r])$. Second, note that

$$\begin{aligned}\mathrm{dist}_B(V(M^{x,y}[l, x]), r) = d_l(x) &\geq \mathrm{dist}_B(x, l) + w(xy) + \mathrm{dist}_B(y, r) - d_r(y)\\ &= w(M^{x,y}) - \kappa = w(M^{x,y}[v, r]).\end{aligned}$$

⌟

Claim 6.32. *Let M be a maximal mountain. Then $M = M^{x,y}$ for some $(x, y) \in Z$.*

Proof. Let κ_M be a real that witnesses that M is a mountain and let $xy \in M$ be such that $w(M[l, x]) \leq \kappa_M \leq w(M[y, r])$ (i.e., the summit of M is on the edge xy, possibly in one of the endpoints). Let $v = v(M, l, \kappa)$. If needed, subdivide the edge xy with the vertex v. By Lemma 6.28 we have that $M[l, x]$ is the leftmost shortest-path between l and x and $M[y, r]$ is the rightmost shortest-path between y and r. By Corollary 6.6, $d_l(x) = \mathrm{dist}_B(V(M[l, x]), r) \geq w(M) - \kappa_M = w(M[y, r]) + w(vy) = \mathrm{dist}_B(y, r) + w(vy)$ and symmetrically $d_r(y) \geq \mathrm{dist}_B(x, l) + w(xv)$. By adding up these two inequalities we obtain $d_r(y) + d_l(x) \geq \mathrm{dist}_B(x, l) + w(xy) + \mathrm{dist}_B(y, r)$. Consequently, $(x, y) \in Z$ and $M^{x,y} = M$ by the construction of $M^{x,y}$. ⌟

By Claims 6.31 and 6.32, our goal is to compute the set of all finite faces that are enclosed by some mountain $M^{x,y}$ for $(x, y) \in Z$.

To achieve this goal, we first construct the directed dual $B^*_\rightarrow$ of B, that is, we take the undirected dual B^* and replace each edge with two arcs in both directions. Then, we would like to assign integer weights to the arcs of $B^*_\rightarrow$ in the following manner. First, set all weights to zero. Second, for each $(x, y) \in Z$, add $+1$ to the weight of each arc that corresponds to an edge of $\partial M^{x,y}$ and ends in the face enclosed by $M^{x,y}$, and add -1 to the weight of the arc in the opposite direction. It is easy to observe that the weighted graph $B^*_\rightarrow$ defined in this manner has no non-null cycles, and for any face f, the sum of weights on any path from the outer face to f in $B^*_\rightarrow$ equals the number of mountains $M^{x,y}$, $(x, y) \in Z$ that enclose f. Consequently, given $B^*_\rightharpoonup$ it is straightforward to compute the union of all finite faces of δ-mountains for fixed l and r.

However, inspecting the perimeters of all mountains $M^{x,y}$ for $(x, y) \in Z$ may take quadratic time. Luckily, one can compute the weights of $B^*_\rightharpoonup$ in $O(|B|)$ time as follows. Start with all weights of $B^*_\rightharpoonup$ set to zero. Then, traverse T_l from the leaves to its root and for each edge $e \in E(T_l)$, compute $\zeta_l(e)$: the number of pairs $(x, y) \in Z$ such that x lies in the tree of $T_l \setminus \{e\}$ that does not contain l. Similarly, compute the values $\zeta_r(e)$ for each $e \in E(T_r)$ that count the number of pairs $(x, y) \in Z$ such that y lies in the tree of $T_r \setminus \{e\}$ that does not contain r. Observe that for each $e \in E(T_l)$, there are exactly $\zeta_l(e)$ mountains $M^{x,y}$ for which e lies on the left slope of $M^{x,y}$. Moreover, in all of these mountains, if we orient e towards the root l of T_l, the face that lies on the left-hand side of e is not enclosed by $M^{x,y}$, and the one that lies on the right-hand side is enclosed by $M^{x,y}$. Hence, we may proceed as follows: for each $e \in E(T_l)$, add weight $\zeta_l(e)$ to the arc of $B^*_\rightharpoonup$ that traverses the edge e, keeping the closer-to-root endpoint of e to the right hand side, and add weight $-\zeta_l(e)$ to the other arc of $B^*_\rightharpoonup$ corresponding to the edge e. Similarly, for each $e \in E(T_r)$, add weight $\zeta_r(e)$ to the arc of $B^*_\rightharpoonup$ that traverses the edge e keeping the closer-to-root endpoint of e to the left hand side, and add weight $-\zeta_r(e)$ to the other arc of $B^*_\rightharpoonup$ corresponding to the edge e. Finally, observe that each mountain $M^{x,y}$ contains the baseline $\partial B[l, r]$ and there are exactly $|Z|$ such mountains. To support this, for each $e \in \partial B[l, r]$, add weight $|Z|$ to all arcs that traverse an edge of $\partial B[l, r]$ and start in the outer face, and add weight $-|Z|$ to such arcs that end in the outer face. In this manner we have constructed the graph $B^*_\rightharpoonup$ in $O(|B|)$ time, and concluded the proof of Lemma 6.30. □

7 Taming sliding trees

In the previous section, we took a major step towards finding a cycle C of length $\mathcal{O}(w(\partial B))$ that lies close to the perimeter of B and that separates the core from all vertices of degree at least three of some optimal solution for any set of terminals on ∂B. In fact, Lemma 6.2 shows that short subtrees of optimal Steiner trees in B are hidden in δ-mountains. Here, 'short' means that the leftmost and rightmost path in the subtree have total length at most $(1/2 - \delta)w(\partial B)$. Note that an optimal Steiner tree in B has total size smaller than $w(\partial B)$, as ∂B without an arbitrary edge connects any subset of $V(\partial B)$. Therefore, for small δ, we can 'hide' almost an entire optimal Steiner tree T in at most two δ-mountains. In this section we study what is left outside these mountains.

Before we describe the main result of this section, we need an additional notion. Let B be an edge-weighted brick. For an edge $uv \in E(B)$ we say that *each point of uv is at distance at most d from $V(\partial B)$* if $uv \in \partial B$ or $\mathrm{dist}_B(u, V(\partial B)) \leq d$, $\mathrm{dist}_B(v, V(\partial B)) \leq d$ and, additionally, $\mathrm{dist}_B(u, V(\partial B)) + \mathrm{dist}_B(v, V(\partial B)) + w(uv) \leq 2d$. Equivalently, we may require that $uv \in \partial B$ or whenever we subdivide the edge uv, replacing it with a new vertex x and edges ux, vx with positive lengths satisfying $w(ux) + w(vx) = w(uv)$, we have $\mathrm{dist}_B(x, V(\partial B)) \leq d$. For a subgraph H of B, we say that *each point of H is at distance at most d from $V(\partial B)$* if each vertex and each point of each edge of H is at distance at most d from $V(\partial B)$.

With this definition, we are ready to state the main theorem of this section.

Theorem 7.1. *Let $\tau \in (0, 1/36]$ be a fixed constant. Assume that B does not admit a short τ-nice tree. Then one can compute a simple cycle C in B with the following properties:*

(i) the length of C is at most $\frac{16}{\tau^2}w(\partial B)$;

(ii) each point of C is within distance at most $(\frac{1}{4} - 2\tau)w(\partial B)$ from $V(\partial B)$;

(iii) for each vertex $x \in V(C)$ there exists a shortest path from x to $V(\partial B)$ such that no edge of the path is strictly enclosed by C;

(iv) C encloses $f_{\mathtt{core}}$, where $f_{\mathtt{core}}$ is any arbitrarily chosen face of B promised by Theorem 5.7 that is not carved by any 2τ-carve;

(v) for any $S \subseteq V(\partial B)$ there exists an optimal Steiner tree T_S connecting S in B such that no vertex of degree at least 3 in T_S is strictly enclosed by C.

The computation takes $\mathcal{O}(|B| \log\log |B|)$ time in the edge-weighted setting and $\mathcal{O}(|B|)$ time in the unweighted setting.

We begin the proof of Theorem 7.1 with a construction. Then we show how it interacts with optimal Steiner trees in B.

Let $\mathbf{P} \subseteq V(\partial B)$ be a set of *pegs* on ∂B, such that for any $v \in V(\partial B)$, there exist pegs $p_{\leftarrow}(v)$ and $p_{\rightarrow}(v)$ with $v \in V(\partial B[p_{\leftarrow}(v), p_{\rightarrow}(v)])$ and $w(\partial B[p_{\leftarrow}(v), v]), w(\partial B[v, p_{\rightarrow}(v)]) \leq \tau w(\partial B)/2$. Here, possibly $p_{\leftarrow}(v) = v$ or $p_{\rightarrow}(v) = v$. We choose the set of pegs $\mathbf{P}$ in the following greedy manner. We take an arbitrary vertex $v_0 \in V(\partial B)$ as a first peg and then we traverse ∂B starting from v_0 twice, once clockwise and once counter-clockwise. In each pass, we take as a next peg the first vertex that is of distance larger than $\tau w(\partial B)/2$ from the previously placed peg. As each pass chooses at most $2/\tau$ pegs, $|\mathbf{P}| \leq 4/\tau$.

Let $\delta = 4\tau$. For any $l, r \in \mathbf{P}$, $l \neq r$, apply Theorem 6.3 to find the mountain range $MR_{l,r}$ for δ-mountains with endpoints l and r. Recall that $MR_{l,r}$ is a set of faces of B. Let $MR = \bigcup_{l,r \in \mathbf{P}, l \neq r} MR_{l,r}$. As $|\mathbf{P}|$ is a constant, by Theorem 6.3 MR is computable within the desired time bound.

Since each δ-mountain is a δ-carve, $f_{\mathtt{core}} \notin MR$. Let $\widehat{f_{\mathtt{core}}}$ be the connected component of $B^* \setminus MR$ containing $f_{\mathtt{core}}$, where B^* is the dual of B without the outer face. Let $C(f_{\mathtt{core}})$ be the simple cycle in B around $\widehat{f_{\mathtt{core}}}$. Clearly, each edge of $C(f_{\mathtt{core}})$ belongs either to some $MR_{l,r} \setminus \partial B$ or to ∂B. Therefore, by Theorem 6.3,

$$w(C(f_{\mathtt{core}})) \leq |\mathbf{P}|(|\mathbf{P}| - 1)(1 - 2\delta)w(\partial B) + w(\partial B) \leq \frac{16}{\tau^2} w(\partial B).$$

Now let B_{close} be the set of edges of B of which each point is at distance at most $(\frac{1}{4} - \frac{\delta}{2})w(\partial B) = (\frac{1}{4} - 2\tau)w(\partial B)$ from $V(\partial B)$; note that B_{close} can be computed in $\mathcal{O}(|B|)$ time by creating a super-terminal vertex t in the outer face of B, connecting it by unit-length edges to all vertices of $V(\partial B)$, and running a shortest-path algorithm from t in the obtained plane graph in linear time [45]. Observe that each edge of $C(f_{\mathtt{core}})$ belongs to B_{close}, since in the definition of $MR_{l,r}$ we consider 4τ-mountains and $\tau \leq \frac{1}{36}$.

Consider now the subgraph H of B that contains all edges of B_{close} that are enclosed by $C(f_{\mathtt{core}})$. Let $f^H_{\mathtt{core}}$ be the face of H that contains $f_{\mathtt{core}}$. As $C(f_{\mathtt{core}})$ is a subgraph of H, $f^H_{\mathtt{core}}$ is a finite face of H. Define C to be some shortest cycle in H separating the outer face of H from $f^H_{\mathtt{core}}$; such a cycle exists as $f^H_{\mathtt{core}}$ is finite. Observe that C corresponds to a minimum cut between $f^H_{\mathtt{core}}$ and the outer face of H in the dual of H. Hence, C can be found in $\mathcal{O}(|B| \log\log |B|)$ time in the edge-weighted setting [47] and in $\mathcal{O}(|B|)$ time in the unweighted setting [31].

We claim that the cycle C satisfies all the requirements of Theorem 7.1. Since $C(f_{\mathtt{core}})$ is a candidate for C, $w(C) \leq w(C(f_{\mathtt{core}})) \leq \frac{16}{\tau^2} w(\partial B)$ and property (i) is satisfied. Properties (ii) and (iv) follows directly from the construction of C.

Regarding property (iii), consider any $x \in V(C)$ and let P_x be a shortest path between x and $V(\partial B)$ that uses the minimum number of edges strictly enclosed by C. Since $x \in V(B_{\text{close}})$, in particular $\mathrm{dist}_B(x, V(\partial B)) \leq (\frac{1}{4} - 2\tau)w(\partial B)$, it is clear that also all edges of P_x are in B_{close}. Assume now that P_x contains some edge strictly enclosed by C. Then P_x contains a subpath P'_x between two vertices $y, z \in V(C)$ that is strictly enclosed by C. By the choice of P_x we infer that $w(C[y, z]), w(C[z, y]) > w(P'_x)$. Since every edge of P'_x is in B_{close}, we infer that either $C[y, z] \cup P'_x$ or $C[z, y] \cup P'_x$ is a cycle that separates $f^H_{\mathtt{core}}$ from the outer face in H of length strictly shorter than $w(C)$, a contradiction to the choice of C. Hence, no edge of P_x is strictly enclosed by C, and property (iii) follows.

The following lemma proves that C satisfies the remaining condition, property (v), and thus finishes the proof of Theorem 7.1.

Lemma 7.2. *For any set $S \subseteq V(\partial B)$ there exists an optimal Steiner tree T_S connecting S in B such that no vertex of degree at least 3 in T_S is strictly enclosed by C.*

Proof. Let T be an optimal Steiner tree in B for some set of terminals; clearly, it is also optimal for the set of terminals $S := V(T) \cap V(\partial B)$. Note that T is a brickable connector and let $\mathcal{B} = \{B_1, B_2, \ldots, B_s\}$ the corresponding brick partition, i.e., $B_1, B_2, \ldots, B_s$ are the bricks induced by the faces of $T \cup \partial B$. Recall that $\sum_{i=1}^{s} w(\partial B_i) \leq w(\partial B) + 2w(T)$.

For each brick B_i, let $a_i, b_i \in V(\partial B)$ be such that $\partial B[a_i, b_i] = \partial B_i \setminus T$. Since T is an optimal Steiner tree for some choice of terminals on ∂B, we have that T is short. By assumption we have that T is not τ-nice, so there exists a brick B_i with $w(\partial B_i) > (1 - \tau)w(\partial B)$. Let B_i be such a large brick. Note that $\partial B[b_i, a_i]$ connects S, so $w(T) \leq w(\partial B[b_i, a_i])$. We infer that

$$\tau w(\partial B) > w(\partial B) - w(\partial B_i) = w(\partial B[b_i, a_i]) - w(\partial B_i \cap T) \geq w(T) - w(\partial B_i \cap T) = w(T \setminus \partial B_i). \quad (1)$$

Note that $\partial B_i = \partial B[a_i, b_i] \cup T[a_i, b_i]$, where by $T[x, y]$ we define the unique path in T between x and y. Let v_a and v_b be vertices on $T[a_i, b_i]$ such that

$$w(T[a_i, v_a]), w(T[b_i, v_b]) \leq \min\left(w(T[a_i, b_i])/2, \left(\frac{1}{2} - 6\tau\right) w(\partial B)\right)$$

and, moreover, both $T[a_i, v_a]$ and $T[b_i, v_b]$ are as long as possible. Note that possibly $v_a = v_b$, but vertices a_i, v_a, v_b, b_i appear on $T[a_i, b_i]$ in this order. In particular, $v_a \neq b_i$ and $v_b \neq a_i$.

Let Z be the union of $\{a_i, b_i\}$ with the set of vertices of $T[a_i, b_i]$ of degree at least 3 in T. Let w_a be the vertex of Z and $T[a_i, v_a]$ that is closest to v_a and let e_a be the edge that precedes $T[w_a, a_i]$ on $T[b_i, a_i]$. Let T_a be the subtree of T rooted at w_a with the parent edge e_a. Note that the rightmost element of $V(T_a) \cap V(\partial B)$ is a_i; let c be the leftmost element of $V(T_a) \cap V(\partial B)$. By (1), $w(T[w_a, c]) \leq \tau w(\partial B)$. Therefore $w(T[w_a, c]) + w(T[w_a, a_i]) \leq (\frac{1}{2} - 5\tau)w(\partial B)$.

Assume that $c \neq a_i$ and $w(\partial B[a_i, c]) \leq w(\partial B)/2$. As $w(T[a_i, c]) \leq (\frac{1}{2} - 5\tau)w(\partial B)$, we infer that $(T[a_i, c], \partial B[a_i, c])$ is a δ-carve and, by Lemma 5.2, $w(\partial B[a_i, c]) \leq (\frac{1}{2} - 4\tau)w(\partial B)$. Let $C := \partial B[a_i, c] \cup T[a_i, c]$, which is a closed walk. Consider the subgraph T' created from T by first deleting any edge enclosed by C, and then adding the closed walk C instead. Note that ∂B_i is enclosed by T' and $w(\partial B_i) > (1 - \tau)w(\partial B)$, thus

$$w(T') \leq w(T) - (1 - \tau)w(\partial B) + (1 - 9\tau)w(\partial B) \leq w(T) - 8\tau w(\partial B).$$

However, as T' includes C, T' also connects S, a contradiction to the choice of T.

Therefore $c = a_i = w_a$ or $w(\partial B[c, a_i]) < w(\partial B)/2$. Consider the second case. Again, we observe that $(T[a_i, c], \partial B[c, a_i])$ is a δ-carve and, by Lemma 5.2, $w(\partial B[c, a_i]) < (\frac{1}{2} - 4\tau)w(\partial B)$. We now use the pegs $p_{\rightarrow}(a_i), p_{\leftarrow}(c) \in \mathbf{P}$. By the choice of $\mathbf{P}$, $w(\partial B[a_i, p_{\rightarrow}(a_i)]) + w(\partial B[p_{\leftarrow}(c), c]) \leq \tau w(\partial B)$. By Lemma 6.2, $((\partial B[p_{\leftarrow}(c), c] \cup T[w_a, c]) \wedge (T[w_a, a_i] \cup \partial B[a_i, p_{\rightarrow}(a_i)]))$ is a δ-mountain and, by Theorem 6.3 and the construction of $C(f_{\mathtt{core}})$, no edge of the subtree of T rooted at

w_a with parent edge e_a is strictly enclosed by $C(f_{\texttt{core}})$, and, hence, by C as well. Clearly, this last claim is also true in the case $c = a_i = w_a$.

Symmetrically, the same argumentation can be made for w_b being the first vertex of Z on $T[v_b, b_i]$, with its preceding edge e_b.

Now, if $T[w_a, w_b]$ does not contain any internal vertex from Z, then every vertex of degree at least 3 in T is contained either in T_a or in T_b, and hence the lemma is proven for $T_S = T$. Therefore, assume otherwise. In particular, by the choice of w_a and w_b, $v_a \neq v_b$, $v_a v_b \notin T$ and $w(T[a_i, b_i]) > (1 - 12\tau)w(\partial B)$. As $\partial B[b_i, a_i]$ connects S, $w(\partial B[b_i, a_i]) \geq w(T[a_i, b_i]) > (1 - 12\tau)w(\partial B)$ and $w(\partial B[a_i, b_i]) < 12\tau w(\partial B)$.

Consider two consecutive vertices w_1, w_2 from Z on $T[a_i, b_i]$. Note that $(T \setminus T[w_1, w_2]) \cup \partial B[a_i, b_i]$ connects S. Therefore, by the minimality of T, $w(T[w_1, w_2]) < 12\tau w(\partial B)$. Recall that $w(T \setminus \partial B_i) \leq \tau w(\partial B)$, and, in particular, any vertex of Z is connected with ∂B with a path in T of length at most $\tau w(\partial B)$. We infer that any edge of $T[a_i, b_i]$ lies on some path of length at most $14\tau w(\partial B)$ with endpoints in $V(\partial B)$ and thus, belongs to B_{close} since $\tau \leq 1/36$.

Let us now take any brick $B_j \neq B_i$. Observe that $w(\partial B_j \cap T) \leq 13\tau w(\partial B)$, since $\partial B_j \cap \partial B_i$ is either empty or an interval of length at most $12\tau w(\partial B)$, and $w(T \setminus \partial B_i) \leq \tau w(\partial B)$. Recall that $\partial B[a_j, b_j] = \partial B_j \setminus T$. Assume first that $w(\partial B[a_j, b_j]) > \frac{1}{2}w(\partial B)$. Observe that then $w(\partial B[b_j, a_j]) \leq \frac{1}{2}w(\partial B)$ and, since $\partial B[b_j, a_j]$ connects S, we would obtain that $w(T) \leq \frac{1}{2}w(\partial B)$ by the optimality of T. On the other hand, $w(T) \geq w(B_i \setminus \partial B[a_i, b_i]) \geq (1 - 13\tau)w(\partial B)$. Since $\tau \leq \frac{1}{36}$, we obtain a contradiction.

Therefore, $w(\partial B[a_j, b_j]) \leq \frac{1}{2}w(\partial B)$. Since $w(T[a_j, b_j]) = w(\partial B_j \cap T) \leq 13\tau w(\partial B)$, $\delta = 4\tau$ and $\tau \leq \frac{1}{36}$, we obtain that $(T[a_j, b_j], \partial B[a_j, b_j])$ is a δ-carve. As a result, we infer that $f_{\texttt{core}}$ is not inside B_j. Since B_j was chosen arbitrarily, $f_{\texttt{core}}$ belongs to B_i.

Assume that some edge of T is strictly enclosed by C. As $f_{\texttt{core}}$ belongs to both B_i and C, this implies that a subpath $T[x, y]$ of $T[a_i, b_i]$ $(x, y \in V(C))$ is strictly enclosed by C. Without loss of generality assume that $T[x, y] \cup C[x, y]$ encloses $f_{\texttt{core}}$, that is, B_i lies on the same side of $T[x, y]$ as $C[x, y]$. Consequently, any edge of T incident to an internal vertex of $T[x, y]$ is enclosed by $T[x, y] \cup C[y, x]$. As each edge of $T[a_i, b_i]$ belongs to B_{close}, by the construction of C we obtain $w(T[x, y]) \geq w(C[y, x])$. Construct T' from T by removing any edge enclosed by $C[y, x] \cup T[x, y]$ and adding $C[y, x]$ instead. Clearly, $w(T') \leq w(T)$, T' connects S and T' contains strictly less edges strictly enclosed by C. By repeating this argument for all subpaths $T[x, y]$, we obtain a subgraph T_S connecting S and without any edge strictly enclosed by C. This finishes the proof of the lemma. □

This concludes the proof of Theorem 7.1.

8 A polynomial kernel: concluding the proof of Theorem 1.1

In this section, we conclude the proof of Theorem 1.1. That is, we assume that the brick B is unweighted.

Fix $\tau = 1/36$ and choose α such that

$$(1 - \tau)^{\alpha - 1} < \tfrac{1}{3}$$

and

$$(1 - 3\tau)^{\alpha - 1} < 1/202177.$$

(In particular, $\alpha > 141$.) We show an algorithm that runs in $\mathcal{O}(|\partial B|^{\alpha}|B|)$ time and returns a subgraph H of size bounded by $\beta|\partial B|^{\alpha}$ for sufficiently large β such that

$$202177(1 - 3\tau)^{\alpha - 1} + 108838883520/\beta \leq 1.$$

For example, $\alpha = 142$ and $\beta = 2\,159\,872\,407\,596$ suffices.

First, consider the base case $|\partial B| \leq 2/\tau = 72$. For each subset $S \subseteq V(\partial B)$, we compute in $\mathcal{O}(|B|)$ time an optimal Steiner tree using the algorithm of Erickson et al. [36] for the set S and add it to graph H. Note that the size of the computed tree is at most 71, as ∂B without an arbitrary edge connects $V(\partial B)$. Therefore, in $\mathcal{O}(|B|)$ time we obtain a graph H of size at most $71 \cdot 2^{72}$, which is at most $\beta|\partial B|^\alpha$ for any $\beta \geq 1$, as $\alpha > 141$ and $|\partial B| \geq 3$.

Now, consider the recursive case. Using the algorithm of Theorem 4.4, we test in $c_1|\partial B|^8 \cdot |B|$ time whether B admits a short τ-nice tree, for some constant c_1. If the algorithm returns a short τ-nice brick covering $\mathcal{B} = \{B_1, B_2, \ldots, B_p\}$, then we recurse on each brick B_i separately, obtaining a subgraph H_i. By Lemma 4.5 and the choice of α, we may return the subgraph $H := \bigcup_{i=1}^p H_i$. As for the time complexity, assume that the i-th recursive call took at most $c|\partial B_i|^\alpha |B_i|$ time. Then, as the brick covering $\mathcal{B}$ is short and τ-nice, we obtain that the total time spent is bounded by

$$\left(c_1|\partial B|^8 + c\sum_{i=1}^p |\partial B_i|^\alpha\right)|B| \leq |\partial B|^\alpha |B| \left(c_1 + 3c(1-\tau)^{\alpha-1}\right),$$

which is at most $c|\partial B|^\alpha |B|$ for sufficiently large c, by the choice of α.

Assume then that the algorithm of Theorem 4.4 decided that no short τ-nice tree exists in B. First, we find some core face $f_{\texttt{core}}$, using Theorem 5.7, that cannot be 2τ-carved. Then we employ Theorem 7.1 to find a cycle C of length at most $\frac{16}{\tau^2}|\partial B| = 20736|\partial B|$ that encloses $f_{\texttt{core}}$. Mark a set $X \subseteq V(C)$ such that the distance between any two consecutive vertices of X on C is at most $2\tau|\partial B| = |\partial B|/18$. As $|\partial B| > 72$, we may greedily mark such set X of size at most $\frac{5}{4}\frac{|C|}{2\tau|\partial B|} \leq 466560$. For each $x \in X$, we compute a shortest path P_x from x to $V(\partial B)$ that does not contain any edge strictly enclosed by C. Note that this computation can be done by a simple breadth-first search from $V(\partial B)$ in the graph obtained from B by removing all edges strictly enclosed by C. Moreover, in this manner, for any $x, y \in X$, the intersection of P_x and P_y is a common (possibly empty) suffix. By condition (ii) of Theorem 7.1, each path P_x is of length at most $(\frac{1}{4} - 2\tau)|\partial B| = \frac{7}{36}|\partial B|$. For $x \in X$, let $\pi(x)$ be the second endpoint of P_x.

Let $x, y \in X$ be two vertices that are consecutive (in counter-clockwise direction) on C and consider the walk $P := P_x \cup C[x,y] \cup P_y$. Note that $|P| \leq \frac{4}{9}|\partial B|$, as $|P_x|, |P_y| \leq \frac{7}{36}|\partial B|$ and $C[x,y] \leq \frac{1}{18}|\partial B|$. We claim that:

$$|\partial B[\pi(x), \pi(y)] \cup P| \leq (1 - 3\tau)|\partial B|. \tag{2}$$

If $\pi(x) = \pi(y)$, then $|\partial B[\pi(x), \pi(y)] \cup P| \leq |P| \leq \frac{4}{9}|\partial B|$, and (2) follows from the choice of τ. Therefore, suppose that $\pi(x) \neq \pi(y)$. Then P_x and P_y do not intersect. Let x' be the vertex of $V(P_x) \cap V(C[x,y])$ that lies closest to $\pi(x)$ on P_y, and define y' similarly with respect to P_y. Observe that x' lies closer to x on $C[x,y]$ than y', as otherwise $P_x[x,x']$ and $P_y[y,y']$ would intersect (recall that neither P_x nor P_y contains an edge strictly enclosed by C). Hence, $C[x',y']$ is a subpath of $C[x,y]$. Define $P' = P_x[\pi(x), x'] \cup C[x',y'] \cup P_y[y', \pi(y)]$. Observe that P' is simple path of length at most $|P| \leq \frac{4}{9}|\partial B|$. Then, either $(P', \partial B[\pi(x), \pi(y)])$ or $(P', \partial B[\pi(y), \pi(x)])$ is a (2τ)-carve. Note that $P' \cup \partial B[\pi(y), \pi(x)]$ encloses C, and thus in particular $f_{\texttt{core}}$. Hence, it must be $(P, \partial B[\pi(x), \pi(y)])$ that is a (2τ)-carve. By Lemma 5.2 we infer that $|\partial B[\pi(x), \pi(y)]| \leq \frac{17}{36}|\partial B|$, and thus $|\partial B[\pi(x), \pi(y)] \cup P| \leq \frac{33}{36}|\partial B|$. Then (2) follows from the choice of τ.

Consider now the closed walk $W_x = \partial B[\pi(x), \pi(y)] \cup P$. Let H_x be the graph consisting of all edges of W_x that neighbour the outer face of W_x treated as a planar graph; note that W_x and H_x are computable in linear time for fixed x. By definition, each doubly-connected component of H_x is a cycle or a bridge. For each doubly-connected component that is a cycle, we create a brick consisting of all edges of B that are enclosed by this cycle. Let $\mathcal{B}_x$ be the family of obtained bricks. Observe that $\mathcal{B}_x$ is computable in linear time and a face of B is enclosed by some brick of $\mathcal{B}_x$ if and only if it is enclosed by W_x. Moreover, by (2),

$$\sum_{B' \in \mathcal{B}_x} |\partial B'| \leq |W_x| \leq (1 - 3\tau)|\partial B|.$$

Therefore,

$$\sum_{x \in X} \sum_{B' \in \mathcal{B}_x} |\partial B'| \leq |C| + |\partial B| + 2|X| \frac{7}{36} |\partial B| \leq 202177 |\partial B|. \tag{3}$$

We recurse on each brick $B' \in \mathcal{B}_x$, obtaining a graph $H(B')$. Furthermore, for each $x, y \in V(C)$, we mark one shortest path $Q_{x,y}$ between x and y in B, if its length is at most $|\partial B|$. We define

$$H := \left(\bigcup_{x \in X} \bigcup_{B' \in \mathcal{B}_x} H(B') \right) \cup \left(\bigcup_{x,y \in X} Q_{x,y} \right).$$

By Theorem 7.1, for any choice of terminals on $V(\partial B)$, there exists an optimal Steiner tree contained in H. Note here that by Theorem 7.1 we may assume that every connection strictly enclosed by C is realized by some marked shortest path $Q_{x,y}$.

We now bound the size of H. For each $x \in X$ and $B' \in \mathcal{B}_x$ we have $|H(B')| \leq \beta |\partial B'|^\alpha$. Moreover, each $Q_{x,y}$ is of length at most $|\partial B|$. Hence,

$$\begin{aligned} |H| &\leq \beta \sum_{x \in X} \sum_{B' \in \mathcal{B}_x} |\partial B'|^\alpha + \binom{|X|}{2} |\partial B| \\ &\leq \beta 202177 |\partial B|^\alpha (1 - 3\tau)^{\alpha - 1} + 108838883520 |\partial B| \\ &\leq \beta |\partial B|^\alpha. \end{aligned}$$

(The last inequality follows from the choice of α and β.)

Regarding time bound, note that all computations, except for the recursive calls, can be done in $c_2 |\partial B|^3 |B|$ time, for some constant c_2. Therefore the total time spent is

$$\left(c_2 |\partial B|^3 + c \sum_{x \in X} |\partial B_x|^\alpha \right) |B| \leq |\partial B|^\alpha |B| \left(c_2 + 202177 c (1 - 3\tau)^{\alpha - 1} \right)$$

which is at most $c|\partial B|^\alpha |B|$ for sufficiently large c, by the choice of α.

9 Dynamic programming to find nice subgraphs

Our goal in this section is to prove the two algorithmic statements mentioned Section 4.

Theorem 9.1 (Theorem 4.4 recalled)**.** *Let $\tau > 0$ be a fixed constant. Given an unweighted brick B, in $\mathcal{O}(|\partial B|^8 |B|)$ time one can either correctly conclude that no short τ-nice tree exists in B or find a short τ-nice brick covering of B.*

Theorem 9.2. *Let $0 < \tau \leq \frac{1}{4}$ be a fixed constant. Given an edge-weighted brick B, in $\mathcal{O}(\tau^{-14} |B| \log |B|)$ time one can either correctly conclude that no 3-short τ-nice tree exists in B or find a $(3+2\tau)$-short $(\tau/2)$-nice brick covering $\mathcal{B}$ of B with the following additional properties:*

1. *each finite face of B is enclosed by at most 7 bricks $B' \in \mathcal{B}$;*

2. $\bigcup_{B' \in \mathcal{B}} \partial B'$ *is connected.*

The idea of the proofs of Theorems 4.4 and 9.2 is to perform a dynamic-programming algorithm similar to the algorithm of Erickson et al. [36] for finding an optimal Steiner tree for a given set of terminals on the outer face. However, as we impose some restrictions on the faces that the tree cuts out of the brick B, the outcome of the algorithm may no longer be a tree. We start by formalizing what we can actually find.

Construct the *extended brick* $\widehat{B}$ as follows: take B and for every $a \in V(\partial B)$ add a degree-1 vertex $\widehat{a}$ attached to a with an edge of zero weight, drawn outside the cycle ∂B. (We remark here that the weight of the edge $a\widehat{a}$ does not have any real significance in the sequel.) We denote $\widehat{\partial B} = \{\widehat{a} : a \in V(\partial B)\}$.

We define an *ordered tree* $\mathbb{T}$ as a rooted tree where every vertex has imposed some linear order on its children. This naturally induces a linear order on the set of leaves of $\mathbb{T}$. The following definition captures the objects found by our dynamic-programming algorithms.

Definition 9.3 (embedded tree)**.** *An* embedded tree *is a pair* $(\mathbb{T}, \pi)$ *where* $\mathbb{T}$ *is an edge-weighted ordered tree with at least one edge, rooted at vertex* $r(\mathbb{T})$*, and* π *is a homomorphism from* $\mathbb{T}$ *into* $\widehat{B}$ *such that* $\pi(v) \in V(B)$ *for any non-leaf vertex of* $\mathbb{T}$ *and* π *assigns the leaves of* $\mathbb{T}$ *to vertices of* $\widehat{\partial B}$*. We require that the order of the leaves of* $\mathbb{T}$ *coincides with the counter-clockwise order of their images on* $\widehat{\partial B}$ *under the homomorphism* π*.*

We say that an embedded tree is leaf-injective *if* π *is injective on the set of leaves of* $\mathbb{T}$*.*

Here, by a homomorphism π from a graph G to a graph H we mean a function $\pi : E(G) \cup V(G) \to E(H) \cup V(H)$ that matches edges to edges and vertices to vertices and, if $\pi(uv) = u'v'$, then $\{\pi(u), \pi(v)\} = \{u', v'\}$ and $w(u'v') = w(uv)$. As all edges $a\widehat{a}$ are of weight zero in $\widehat{B}$, and all edges of B have positive weight, we may restrict ourselves to embedded trees where an edge has weight zero if and only if it is adjacent to a leaf.

We measure the length of an embedded tree as in all weighted graphs. Note that the edges incident to leaves of an embedded tree do not contribute to the length of the tree. In the unweighted case, we will mostly be working with leaf-injective embedded trees, while in the weighted case it will be more convenient to drop this assumption.

Recall that for two vertices $a, b \in \partial B$, by $\partial B[a, b]$ we denote the subpath of ∂B between a and b, obtained by traversing ∂B in counter-clockwise direction. If $a = b$, then $\partial B[a, b] = \emptyset$. We define $\partial^{\uparrow} B[a, b]$ to be equal $\partial B[a, b]$ unless $a = b$; in this case $\partial^{\uparrow} B[a, b] = \partial B$.

An embedded tree $(\mathbb{T}, \pi)$ is τ*-nice* if for any two consecutive leaves $\widehat{l_a}, \widehat{l_b}$ in $\mathbb{T}$ the following holds. Let $\pi(\widehat{l_a}) = \widehat{a}$ and $\pi(\widehat{l_b}) = \widehat{b}$ and let l_a, l_b be the parents of $\widehat{l_a}, \widehat{l_b}$ in $\mathbb{T}$, respectively; note that $\pi(l_a) = a$ and $\pi(l_b) = b$, and possibly $a = b$. Let u be the lowest common ancestor of $\widehat{l_a}$ and $\widehat{l_b}$ in $\mathbb{T}$. Then for $(\mathbb{T}, \pi)$ to be τ*-nice* we require that

$$w(\partial B[a, b]) + w(\mathbb{T}[u, l_a]) + w(\mathbb{T}[u, l_b]) \leq (1 - \tau) w(\partial B). \tag{4}$$

An embedded tree is *fully* τ*-nice* if additionally (4) holds for $\widehat{l_a}$ being the last leaf of $\mathbb{T}$, $\widehat{l_b}$ being the first leaf of $\mathbb{T}$, $u = r(\mathbb{T})$ and $\partial B[a, b]$ replaced by $\partial^{\uparrow} B[a, b]$.

The intuition behind this notion is that the image of $\mathbb{T}[w, l_a] \cup \mathbb{T}[w, l_b]$ under π, together with $\partial B[a, b]$ (or $\partial^{\uparrow} B[a, b]$ in the case of the last and the first leaf of $\mathbb{T}$), is likely to yield a perimeter of an output brick B_i in our algorithm.

We now formalize how to find a set of bricks promised by Theorem 4.4 and Theorem 9.2, given a fully τ-nice embedded tree.

Lemma 9.4. *Given a fully* τ*-nice embedded tree* $(\mathbb{T}, \pi)$ *with* r *leaves, one can in* $\mathcal{O}(r(|\mathbb{T}| + |B|))$ *time compute a* τ*-nice brick covering* $\mathcal{B}$ *of* B *of total perimeter at most* $w(\partial B) + 2w(\mathbb{T})$ *with the following additional properties:*

1. *each finite face of B is enclosed by at most r bricks of $\mathcal{B}$;*
2. *$\bigcup_{B' \in \mathcal{B}} \partial B'$ is connected.*

Proof. Let $\mathcal{F}$ be a family of pairs of two consecutive leaves of $\mathbb{T}$ and the pair $\mathbf{p}^\circ$ consisting of the last and the first leaf of $\mathbb{T}$. For any $\mathbf{p} = (\widehat{l_a}, \widehat{l_b}) \in \mathcal{F}$, define l_a, l_b, a, b, w as in the definition of a fully τ-nice tree. Define $C(\mathbf{p}) := \pi(\mathbb{T}[l_a, w] \cup \mathbb{T}[w, l_b]) \cup \partial B[a, b]$ if $\mathbf{p} \neq \mathbf{p}^\circ$ and $C(\mathbf{p}) := \pi(\mathbb{T}[l_a, w] \cup \mathbb{T}[w, l_b]) \cup \partial^\uparrow B[a, b]$ if $\mathbf{p} = \mathbf{p}^\circ$. Observe that $C(\mathbf{p})$ is a closed walk in B. Note that the fact that $\mathbb{T}$ is fully τ-nice tree implies that the length of $C(\mathbf{p})$ is bounded by $(1 - \tau)w(\partial B)$. Moreover, as edges of $\mathbb{T}$ not incident to a leaf contribute to exactly two cycles $C(\mathbf{p})$, and each edge of ∂B contributes to exactly one such cycle, we have

$$\sum_{\mathbf{p} \in \mathcal{F}} w(C(\mathbf{p})) = w(\partial B) + 2w(\mathbb{T}). \tag{5}$$

Let $H_0(\mathbf{p})$ be the subgraph of B consisting of all edges that lie on $C(\mathbf{p})$. Clearly, $H_0(\mathbf{p})$ is connected. Let $H(\mathbf{p})$ be the subgraph of $H_0(\mathbf{p})$ consisting of all edges of $H_0(\mathbf{p})$ that are adjacent to the outer face of $H_0(\mathbf{p})$. Note that $H(\mathbf{p})$ is connected, $\partial B[a, b] \subseteq H(\mathbf{p})$ ($\partial^\uparrow B[a, b] \subseteq H(\mathbf{p})$ if $\mathbf{p} \neq \mathbf{p}^\circ$) and the outer faces of $H(\mathbf{p})$ and $H_0(\mathbf{p})$ are equal. Moreover, by the definition of $H(\mathbf{p})$, any doubly-connected component of $H(\mathbf{p})$ is either a simple cycle or a bridge.

We construct a preliminary brick covering $\mathcal{B}_0$ as follows: for each $\mathbf{p} \in \mathcal{F}$ and for each doubly-connected component D of $H(\mathbf{p})$ that is a cycle, we insert into $\mathcal{B}_0$ a brick B_i consisting of all edges of B that are enclosed by D; clearly $\partial B_i = D$ and B_i is a subbrick of B. Note that $\mathcal{B}_0$ can be computed within the desired running time. Indeed, $H(\mathbf{p})$ can be computed in $\mathcal{O}(|\mathbb{T}| + |B|)$ time, and the corresponding bricks can be computed in $\mathcal{O}(|B|)$ time. It remains to observe that $|\mathcal{F}| = r$, where r is the number of leaves of $\mathbb{T}$.

We can now make several observations about $\mathcal{B}_0$. First, as $w(C(\mathbf{p})) \leq (1 - \tau)w(\partial B)$, each brick in $\mathcal{B}_0$ has perimeter at most $(1 - \tau)w(\partial B)$. Second, for a fixed $\mathbf{p}$, the total perimeter of the bricks inserted into $\mathcal{B}_0$ is at most $w(H(\mathbf{p})) \leq w(C(\mathbf{p}))$. Therefore, by (5), the sum of the perimeters of all bricks in $\mathcal{B}_0$ is bounded by $w(\partial B) + 2w(\mathbb{T})$, as desired. Third, for a fixed cycle $C(\mathbf{p})$, the constructed bricks do not share an enclosed finite face of B. Hence, each finite face of B is enclosed by at most r bricks of $\mathcal{B}_0$.

We now show that $\mathcal{B}_0$ is a brick covering of B, that is, we prove that each face of B is contained in some brick of $\mathcal{B}_0$. Let f be any face of B and let c be an arbitrary point of the plane in the interior of f. Let $\Gamma \cong \mathbb{Z}$ be the fundamental group of $\Pi \setminus \{c\}$, and let ι be the mapping that assigns to each closed curve in $\Pi \setminus \{c\}$ the corresponding element of Γ. For each $\mathbf{p} \in \mathcal{F}$, orient the walk $C(\mathbf{p})$ in the direction such that the part $\partial B[a, b]$ or $\partial^\uparrow B[a, b]$ is traversed from a to b (note that if $\mathbf{p} = \mathbf{p}^\circ$, then $a \neq b$ and $\partial^\uparrow B[a, b] = \partial B[a, b]$, as $(\mathbb{T}, \pi)$ is fully τ-nice). If c belongs to the outer face of the graph $H(\mathbf{p})$, then $C(\mathbf{p})$ is continuously retractable to a single point in $\Pi \setminus \{c\}$, and thus $\iota(C(\mathbf{p}))$ is the neutral element of Γ. On the other hand, $\iota(\partial B)$ is *not* the neutral element of this fundamental group, since it winds around c exactly one time. Observe that in this fundamental group we have equation

$$\sum_{\mathbf{p} \in \mathcal{F}} \iota(C(\mathbf{p})) = \iota(\partial B),$$

since for each $e \in E(\mathbb{T})$ we have that $\pi(e)$ is traversed by two different walks $C(\mathbf{p}_1)$, $C(\mathbf{p}_2)$, in different directions. Therefore, for at least one $\mathbf{p}_0 \in \mathcal{F}$ it must hold that $\iota(C(\mathbf{p}_0))$ is not the neutral element of Γ. Consequently, c belongs to some bounded face of one of the constructed graphs $H(\mathbf{p})$, and one of the bricks of $\mathcal{B}_0$ contains f.

Observe that $\mathcal{B}_0$ has all the required properties, except possibly the property that $\bigcup_{B' \in \mathcal{B}_0} \partial B'$ is connected. To ensure this property as well, we select a subfamily of $\mathcal{B}_0$ as follows. For each connected component D of $\bigcup_{B' \in \mathcal{B}_0} \partial B'$, let $\mathcal{B}_D$ be the family of all bricks $B' \in \mathcal{B}_0$ with $\partial B' \subseteq D$. Let D_0 be the component of $\bigcup_{B' \in \mathcal{B}_0} \partial B'$ that contains ∂B.

We claim that if $D \neq D_0$ is a component of $\bigcup_{B' \in \mathcal{B}_0} \partial B'$, then $\mathcal{B}_0 \setminus \mathcal{B}_D$ is a brick covering of B as well. Let f be a face of B that is incident to one of the edges of D, but is contained in the outer face of D. As D does not contain any edge of ∂B, f is finite. Let $B' \in \mathcal{B}_0$ be a brick such that $\partial B'$ encloses f. Clearly, $B' \notin \mathcal{B}_D$ and hence $\partial B'$ does not share any vertex with D. As f is incident with an edge of D, we infer that $\partial B'$ strictly encloses all edges of D; in particular, $\partial B'$ encloses all faces that are enclosed by the bricks of $\mathcal{B}_D$. Consequently, $\mathcal{B}_0 \setminus \mathcal{B}_D$ is a brick covering of B.

We now remove all bricks $\mathcal{B}_D$ from $\mathcal{B}_0$ for any component $D \neq D_0$ of $\bigcup_{B' \in \mathcal{B}_0} \partial B'$. By the above claim, we infer that the remainder, $\mathcal{B}_{D_0}$, is a brick covering of B. As $\mathcal{B}_{D_0} \subseteq \mathcal{B}_0$, $\mathcal{B}_{D_0}$ inherited all other required properties: in particular, it is τ-nice and of total perimeter at most $w(\partial B) + 2w(\mathbb{T})$. Hence, the algorithm may output $\mathcal{B}_{D_0}$. Observe that it can be computed from $\mathcal{B}_0$ in time linear in $|B|$ and the total size of $\mathcal{B}_0$. □

In the other direction, it is easy to see that a short τ-nice tree in B yields a fully τ-nice embedded tree of small length.

Lemma 9.5. *If B admits a τ-nice tree T, then B admits a fully τ-nice, leaf-injective embedded tree $(\mathbb{T}, \pi)$ of length $w(T)$.*

Proof. We construct $\mathbb{T}$ as follows: root T at an arbitrary vertex $r \in V(T)$, for each $a \in V(T) \cap V(\partial B)$, add the edge $a\widehat{a}$, and for each internal vertex p of $\mathbb{T}$, order its children in the counter-clockwise order in which they appear on the plane (starting from the parent of p, or at arbitrary point for $r = p$). As each leaf of T lies on $V(\partial B)$, in this manner each leaf of $\mathbb{T}$ lies in $\widehat{\partial B}$. Therefore, if we take π to be the identity mapping, $(\mathbb{T}, \pi)$ is an embedded tree. By construction, $w(\mathbb{T}) = w(T)$ and $(\mathbb{T}, \pi)$ is leaf-injective. Moreover, for any two consecutive leaves $\widehat{a}$ and $\widehat{b}$ of $\mathbb{T}$, if w is the lowest common ancestor of $\widehat{a}$ and $\widehat{b}$, then the value $\mathbb{T}[w, a] \cup \mathbb{T}[w, b] \cup \partial B[a, b]$ is the perimeter of the face of $B[T \cup \partial B]$ that neighbours $\partial B[a, b]$. As T is τ-nice, we infer that $(\mathbb{T}, \pi)$ is τ-nice as well. Finally, if $\widehat{a}$ is the last leaf of $\mathbb{T}$ and $\widehat{b}$ is the first leaf of $\mathbb{T}$, then since r has degree at least two in $\mathbb{T}$, r is the lowest common ancestor of $\widehat{a}$ and $\widehat{b}$ in $\mathbb{T}$, and $\mathbb{T}[r, a] \cup \mathbb{T}[r, b] \cup \partial^{\uparrow} B[a, b]$ is again the perimeter of the face of $B[T \cup \partial B]$ that neighbours $\partial^{\uparrow} B[a, b]$. We infer that $(\mathbb{T}, \pi)$ is fully τ-nice and the lemma is proven. □

By Lemmata 9.4 and 9.5, it remains to find a fully τ-nice embedded tree of small length. Here the argumentation for the unweighted and the edge-weighted cases diverge. In both cases, we use a dynamic-programming algorithm. However, in the unweighted case we are able to obtain the exact statement of Theorem 4.4; in the edge-weighted case, we need to perform some rounding to fit into the $\mathcal{O}(|B| \log |B|)$ time frame, and therefore we may lose some 'niceness' of the constructed tree.

9.1 Finding a nice embedded tree in the unweighted setting

For brevity we denote $n = |B|$ and $k = |\partial B|$.

Lemma 9.6. *Assume B is unweighted. Given an integer ℓ, in $\mathcal{O}(nk^4\ell^4)$ time one can find a fully τ-nice leaf-injective embedded tree of length at most ℓ or correctly conclude that no such tree exists.*

Proof. For each $v \in V(B)$, $a, b \in V(\partial B)$, $0 \leq k_a, k_b \leq \ell$, we define $\mathcal{F}[v, a, b, k_a, k_b]$ to be the set of all leaf-injective embedded trees $(\mathbb{T}, \pi)$ that:

1. have length at most ℓ;
2. are τ-nice;
3. satisfy $\pi(r(\mathbb{T})) = v$;
4. map the first leaf of $\mathbb{T}$, $\widehat{l_a}$, to $\widehat{a}$ under π, and the last leaf of $\mathbb{T}$, $\widehat{l_b}$, to $\widehat{b}$ under π;
5. satisfy $|\mathbb{T}[r(\mathbb{T}), l_a]| \leq k_a$ and $|\mathbb{T}[r(\mathbb{T}), l_b]| \leq k_b$, where l_a, l_b are parents of $\widehat{l_a}$, $\widehat{l_b}$ in $\mathbb{T}$, respectively.

Let $M[v, a, b, k_a, k_b] = \min\{w(\mathbb{T}) : (\mathbb{T}, \pi) \in \mathcal{F}[v, a, b, k_a, k_b]\}$.

Assume that B admits a fully τ-nice leaf-injective embedded tree $(\mathbb{T}, \pi)$ of length at most ℓ. Let $\widehat{l_a}$, $\widehat{l_b}$ be the first and the last leaf of $\mathbb{T}$, let l_a, l_b be the parents of $\widehat{l_a}, \widehat{l_b}$ in $\mathbb{T}$, respectively, and let $a = \pi(l_a)$, $b = \pi(l_b)$. Note that $(\mathbb{T}, \pi) \in \mathcal{F}[r(\mathbb{T}), a, b, |\mathbb{T}[r(\mathbb{T}), l_a]|, |\mathbb{T}[r(\mathbb{T}), l_b]|]$ and $|\mathbb{T}[r(\mathbb{T}), l_a]| + |\mathbb{T}[r(\mathbb{T}), l_b]| + |\partial^{\uparrow} B[b, a]| \leq (1-\tau)k$, as $(\mathbb{T}, \pi)$ is fully τ-nice. In the other direction, if $(\mathbb{T}, \pi) \in \mathcal{F}[v, a, b, k_a, k_b]$ and $k_a + k_b + |\partial^{\uparrow} B[b, a]| \leq (1-\tau)k$, then $(\mathbb{T}, \pi)$ is fully τ-nice. Therefore, it suffices to compute, for each choice of the parameters v, a, b, k_a, k_b, the value $M[v, a, b, k_a, k_b]$ and one representative element $T[v, a, b, k_a, k_b] \in \mathcal{F}[v, a, b, k_a, k_b]$ of length $M[v, a, b, k_a, k_b]$, if $\mathcal{F}[v, a, b, k_a, k_b] \neq \emptyset$.

Clearly, for $v = a = b$, $M[v, a, b, k_a, k_b] = 0$ and $T[v, a, b, k_a, k_b]$ can be defined as a two-vertex tree with root r, mapped to $v = a = b$, and a single leaf mapped to $\widehat{a} = \widehat{b}$. These are the only embedded trees of zero length.

Consider a τ-nice leaf-injective embedded tree $(\mathbb{T}, \pi)$ with $0 < w(\mathbb{T}) \leq \ell$. Let $\widehat{l_a}, \widehat{l_b}$ be the first and the last leaf of $\mathbb{T}$, let l_a, l_b be the parents of $\widehat{l_a}, \widehat{l_b}$ in $\mathbb{T}$ and let $a = \pi(l_a)$, $b = \pi(l_b)$, $v = \pi(r(\mathbb{T}))$. Let k_a, k_b be such that $|\mathbb{T}[r(\mathbb{T}), l_a]| \leq k_a$ and $|\mathbb{T}[r(\mathbb{T}), l_b]| \leq k_b$. Consider two cases: either $r(\mathbb{T})$ is of degree one in $\mathbb{T}$ or larger.

In the first case, let p be the only child of $r(\mathbb{T})$; note that p is not a leaf as $w(\mathbb{T}) > 0$. Let $\mathbb{T}_1 = \mathbb{T} \setminus r(\mathbb{T})$, rooted at p, and let π_1 be the mapping π restricted to $\mathbb{T}_1$. Clearly, $(\mathbb{T}_1, \pi_1)$ is a τ-nice embedded tree that belongs to $\mathcal{F}[\pi(p), a, b, k_a - 1, k_b - 1]$.

In the other direction, consider the cell $\mathcal{F}[v, a, b, k_a, k_b]$. We note that for any $w \in N_B(v)$ and $(\mathbb{T}_1, \pi_1) \in \mathcal{F}[w, a, b, k_a - 1, k_b - 1]$, if we extend $\mathbb{T}_1$ with a new root vertex r mapped to v, with one child $r(\mathbb{T}_1)$, then the extended tree belongs to $\mathcal{F}[v, a, b, k_a, k_b]$.

In the second case, split $\mathbb{T}$ into two trees $\mathbb{T}_1$ and $\mathbb{T}_2$, rooted at $r(\mathbb{T})$: $\mathbb{T}_1$ contains the subtree of $\mathbb{T}$ rooted in the first child of $r(\mathbb{T})$, together with the edge connecting it to $r(\mathbb{T})$, and $\mathbb{T}_2$ contains the remaining edges of $\mathbb{T}$ (i.e., all but the first children of $r(\mathbb{T})$, together with the edges connecting them to $r(\mathbb{T})$). Define π_1 and π_2 as restrictions of π to $\mathbb{T}_1$ and $\mathbb{T}_2$, respectively. Let $\widehat{l_c}$ be the last leaf of $\mathbb{T}_1$ and $\widehat{l_d}$ be the first leaf of $\mathbb{T}_2$. Define l_c, l_d, c, d analogously to l_a, l_b, a, b. Observe that $l_c \neq l_d$ since $(\mathbb{T}, \pi)$ is leaf-injective, but it may be that $l_a = l_c$ or $l_d = l_b$ in case $a = c$ or $b = d$. Note that $(\mathbb{T}_1, \pi_1) \in \mathcal{F}[v, a, c, k_a, |\mathbb{T}[r(\mathbb{T}), l_c]|]$ and $(\mathbb{T}_2, \pi_2) \in \mathcal{F}[v, d, b, |\mathbb{T}[r(\mathbb{T}), l_d]|, k_b]$. Moreover, $|\mathbb{T}[r(\mathbb{T}), l_c]| + |\mathbb{T}[r(\mathbb{T}), l_d]| + |\partial B[c, d]| \leq (1 - \tau)k$, as $\mathbb{T}$ is τ-nice and $r(\mathbb{T})$ is the lowest common ancestor of l_c and l_d in $\mathbb{T}$.

In the other direction, assume that for some $c, d \in \partial B[a, b]$ such that c lies strictly closer to a than d on $\partial B[a, b]$ (i.e., $\partial B[a, c] \subsetneq \partial B[a, d] \subseteq \partial B[a, b]$), and for some $k_c, k_d \leq \ell$ such that $k_c + k_d + |\partial B[c, d]| \leq (1 - \tau)k$, we have embedded trees $(\mathbb{T}_1, \pi_1) \in \mathcal{F}[v, a, c, k_a, k_c]$ and $(\mathbb{T}_2, \pi_2) \in \mathcal{F}[v, d, b, k_d, k_b]$ such that $w(\mathbb{T}_1) + w(\mathbb{T}_2) \leq \ell$. Define $\mathbb{T}$ as $\mathbb{T}_1 \cup \mathbb{T}_2$ with identified roots, rooted at $r(\mathbb{T}) = r(\mathbb{T}_1) = r(\mathbb{T}_2)$, and order of the children of $r(\mathbb{T})$ by first placing the children in $\mathbb{T}_1$ and then the children in $\mathbb{T}_2$, in the corresponding orders. Moreover, define $\pi = \pi_1 \cup \pi_2$. Then in the embedded tree $(\mathbb{T}, \pi)$ the first leaf is $\widehat{l_a}$ with $\pi(\widehat{l_a}) = \widehat{a}$ and the last leaf is $\widehat{l_b}$ with $\pi(\widehat{l_b}) = \widehat{b}$. The assumption that c is strictly closer to a than d implies that

$(\mathbb{T},\pi)$ is leaf-injective. Furthermore, $w(\mathbb{T}) = w(\mathbb{T}_1) + w(\mathbb{T}_2) \le \ell$. Finally, the requirement $k_c + k_d + |\partial B[c,d]| \le (1-\tau)k$ implies that $(\mathbb{T},\pi)$ is τ-nice. Hence, $(\mathbb{T},\pi) \in \mathcal{F}[v,a,b,k_a,k_b]$.

From the previous discussion, we infer that $M[v,a,b,k_a,k_b] = 0$ if $v = a = b$ and otherwise $M[v,a,b,k_a,k_b]$ equals the minimum over the following candidates:

- if $k_a, k_b > 0$, for each $w \in N_B(v)$, we take $1 + M[w,a,b,k_a-1,k_b-1]$ as a candidate value;
- for each $c, d \in \partial B[a,b]$ such that c lies strictly closer to a than d on $\partial B[a,b]$, and for each integers $0 \le k_c, k_d \le \ell$ such that $k_c + k_d + |\partial B[c,d]| \le (1-\tau)k$, we take $M[v,a,c,k_a,k_c] + M[v,d,b,k_d,k_b]$ as a candidate value, provided that this value does not exceed ℓ.

We note that, in the aforementioned recursive formula, to compute $M[v,a,b,k_a,k_b]$ we take into account at most $|N_B(v)| + k^2(1+\ell)^2$ other candidates, in each computation taking into account values $M[v',a',b',k'_a,k'_b]$ with $|\partial B[a',b']|$ strictly smaller than $|\partial B[a,b]|$. We infer that the values $M[v,a,b,k_a,k_b]$ for all valid choice of the parameters v,a,b,k_a,k_b can be computed in $\mathcal{O}(nk^4\ell^4)$ time. If we additionally store for each cell $M[v,a,b,k_a,k_b]$ which candidate attained the minimum value, we can read an optimal embedded tree $T[v,a,b,k_a,k_b]$ in linear time with respect to its size. This concludes the proof of the lemma. □

We may now conclude the proof of Theorem 4.4. Using Lemma 9.6 we look for a fully τ-nice embedded tree of length at most k. If one is found, we apply Lemma 9.4 to obtain the desired family of bricks. If the algorithm of Lemma 9.6 does not find any embedded tree, Lemma 9.5 allows us to conclude that no short τ-nice tree exists in B.

9.2 Finding a nice embedded tree in the edge-weighted setting

We start with the following observation that extends Lemma 9.5.

Lemma 9.7. *Let B be an edge-weighted brick and let $0 < \tau \le \frac{1}{4}$ be a constant. If there exists a short τ-nice tree T in B, then there exists an embedded fully τ-nice tree $(\mathbb{T},\pi)$ in B of length at most $w(T)$ and with at most 7 leaves.*

Proof. Let T be as in the lemma statement, and construct $(\mathbb{T},\pi)$ as in the proof of Lemma 9.5. That is, we construct $\mathbb{T}$ as follows: we root T at an arbitrary vertex $r \in V(T)$, for each $a \in V(T) \cap V(\partial B)$, add the edge $a\widehat{a}$, and for each internal vertex p of $\mathbb{T}$, order its children in the counter-clockwise order in which they appear on the plane (starting from the parent of p, or at arbitrary point for $r = p$). The mapping π is the identity mapping. Clearly, $w(\mathbb{T}) = w(T)$. Our goal is to trim $\mathbb{T}$ so that it is still fully τ-nice, but has at most 7 leaves.

Assume $\mathbb{T}$ has at least 8 leaves, as otherwise we are done. Pick any four pairwise distinct leaves $\widehat{l_1}, \widehat{l_2}, \widehat{l_3}, \widehat{l_4}$ of $\mathbb{T}$ with the following properties: they lie in $\mathbb{T}$ in this order, no two of them are two consecutive leaves of $\mathbb{T}$, and $\widehat{l_4}$ is not the last leaf of $\mathbb{T}$. As $\mathbb{T}$ has at least 8 leaves, this is always possible (e.g., we may take the first, third, fifth and seventh leaf of $\mathbb{T}$). Let l_i be the unique neighbour of $\widehat{l_i}$ in $\mathbb{T}$. Moreover, let $\widehat{a_i} = \pi(\widehat{l_i})$ and $a_i = \pi(l_i)$; note that a_i is the unique neighbour of $\widehat{a_i}$ in the extended brick $\widehat{B}$. We use a cyclic ordering for the index i, that is $l_5 = l_1$, $a_5 = a_1$ etc. Observe that all a_i are pairwise distinct, as we have started from a short τ-nice tree T (in other words, $(\mathbb{T},\pi)$ is leaf-injective).

For $i = 1,2,3,4$, by L_i we denote the set of leaves of $\mathbb{T}$ that lie between $\widehat{l_i}$ and $\widehat{l_{i+1}}$ (exclusive), in the circular order of the leaves of $\mathbb{T}$. By the assumption on the leaves $\widehat{l_i}$, all sets L_i are nonempty. For $i = 1,2,3,4$, let $\mathbb{T}_i$ be a subtree of $\mathbb{T}$ defined as follows: for each $\widehat{l} \in L_i$, we remove from $\mathbb{T}$ the path from $\widehat{l}$ to the closest vertex of $\mathbb{T}[\widehat{l_i}, \widehat{l_{i+1}}]$ (recall that $\widehat{l_5} = \widehat{l_1}$). Define $\pi_i = \pi|_{\mathbb{T}_i}$. As we preserve the path $\mathbb{T}[\widehat{l_i}, \widehat{l_{i+1}}]$ in $\mathbb{T}_i$, no new leaf has been introduced into $\mathbb{T}_i$ and $(\mathbb{T}_i, \pi_i)$ is an embedded tree in B.

We claim that for at least one index i, the embedded tree $(\mathbb{T}_i, \pi_i)$ is fully τ-nice. Assume the contrary. Since $(\mathbb{T}, \pi)$ is fully τ-nice, we infer that for each $i = 1, 2, 3, 4$:

$$w(\mathbb{T}[l_i, l_{i+1}]) + w(\partial B[a_i, a_{i+1}]) > (1 - \tau)w(\partial B).$$

Summing up, we infer that:

$$w(\partial B) + \sum_{i=1}^{4} w(\mathbb{T}[l_i, l_{i+1}]) > 4(1 - \tau)w(\partial B) \geq 3w(\partial B),$$

where the last inequality follows from the assumption $\tau \leq \frac{1}{4}$. However, note that

$$\sum_{i=1}^{4} w(\mathbb{T}[l_i, l_{i+1}]) \leq 2w(\mathbb{T}) \leq 2w(T) \leq 2w(\partial B),$$

since T is short. We have reached a contradiction.

Consequently, we may replace $(\mathbb{T}, \pi)$ with $(\mathbb{T}_i, \pi_i)$ for some $i \in \{1, 2, 3, 4\}$, keeping the fully τ-niceness and decreasing the number of leaves. If we proceed with this procedure exhaustively, we finally arrive at an embedded tree that is fully τ-nice and has at most 7 leaves. □

A *branching vertex* is a vertex of an embedded tree $(\mathbb{T}, \pi)$ with at least two children. By Lemma 9.7, in the case $\tau \leq \frac{1}{4}$ we may look for embedded trees with at most 7 leaves and, consequently, at most 6 branching vertices. If we are satisfied with any polynomial running time of the algorithm that finds a fully τ-nice embedded tree, observe that it suffices to guess the images of all leaves and branching vertices of the tree in question, and compute a shortest path between any pair of them. However, if we aim for a $\mathcal{O}(\tau^{-14}|B| \log |B|)$ running time, then we need to proceed more carefully. We will essentially follow the dynamic-programming algorithm of the unweighted case (i.e., Lemma 9.6) but due to the existence of arbitrary real weights, we cannot directly use k_a and k_b, the lengths of the leftmost and rightmost paths in the constructed tree, as dimensions in the dynamic programming table. Instead, we need to round them. The idea is to round independently the length of each maximal path consisting of vertices of degree two of the embedded tree in question; as there are at most 13 such paths, we control the error introduced by the rounding.

Lemma 9.8. *In $\mathcal{O}(\tau^{-14}|B| \log |B|)$ time one can either correctly conclude that no fully τ-nice embedded tree with at most 7 leaves and of length at most $w(\partial B)$ exists in B, or find a fully $(\tau/2)$-nice embedded tree in B of length at most $(1 + \tau)w(\partial B)$.*

Proof. Greedily, we find a set $\mathbf{P} \subseteq V(B)$ of at most $16/\tau$ *pegs*, such that for any $v \in V(\partial B)$, if we traverse ∂B from v in clockwise direction, then we encounter a peg at distance at most $\tau w(\partial B)/8$ (possibly, the peg is on v). Observe the following:

Claim 9.9. *If there exists in B a fully τ-nice embedded tree with at most 7 leaves and of length at most $w(\partial B)$, then there exists a fully $(3\tau/4)$-nice embedded tree with at most 7 leaves and of length at most $(1 + 7\tau/8)w(\partial B)$, whose leaves are mapped to vertices of $\widehat{\partial B}$ adjacent to pegs.*

Proof. Let $(\mathbb{T}, \pi)$ be an embedded tree as in the statement. For each leaf $\widehat{l_a}$ of $\mathbb{T}$, proceed as follows. Let l_a be the unique neighbour of $\widehat{l_a}$ in $\mathbb{T}$ and let $\pi(\widehat{l_a}) = \widehat{a}$ and $\pi(l_a) = a$. Traverse ∂B from a in clockwise direction and let $p(a)$ be the first peg encountered (possibly, $p(a) = a$). Replace the edge $\widehat{l_a}l_a$ in $\mathbb{T}$ with a copy of the path $\partial B[p(a), a]$ and the edge $\widehat{p(a)}p(a)$, embedded by π into $\partial B[p(a), a] \cup \{\widehat{p(a)}p(a)\}$. Note that the constructed tree is an embedded tree. As $w(\partial B[p(a), a]) \leq \tau w(\partial B)/8$, the constructed tree is fully $(\tau - \tau/4)$-nice and we have enlarged the length of $\mathbb{T}$ by at most $7\tau w(\partial B)/8$. ⌟

Hence, we restrict ourselves to embedded trees whose leaves are mapped to the neighbours of pegs. We branch into $\mathcal{O}(|\mathbf{P}|^7) = \mathcal{O}(\tau^{-7})$ cases, guessing the number of leaves and their images in the tree in question. That is, we are now given an integer $r \leq 7$ and a sequence $a_1, a_2, \ldots, a_r$ of pegs that appear on ∂B in this counter-clockwise order (possibly $a_i = a_{i+1}$ for some i), and we look for a fully $(3\tau/4)$-nice embedded tree of length at most $(1 + 7\tau/8)w(\partial B)$ with r leaves that maps consecutive leaves to vertices $\widehat{a_1}, \widehat{a_2}, \ldots, \widehat{a_r}$.

Denote $\lambda = \frac{\tau}{104} w(\partial B)$. As discussed earlier, to achieve the promised running time, we need to round the distances in the dynamic programming algorithm. We will use λ as one unit of distance for rounding. For a real x, by $\mathtt{rnd}(x)$ we denote the smallest integer k for which $k\lambda \geq x$, that is, $\mathtt{rnd}(x) = \lceil x/\lambda \rceil$. For an embedded tree $(\mathbb{T}, \pi)$ with ρ leaves, by $\mathbf{I}(\mathbb{T})$ we denote the set consisting of the root, all branching vertices, and all neighbours of leaves in the tree $\mathbb{T}$. Observe that $|\mathbf{I}(\mathbb{T})| \leq 2\rho$. Let $\mathbb{T}' \subseteq \mathbb{T}$ be any subtree of $\mathbb{T}$. The set $\mathbf{I}(\mathbb{T})$ partitions the edge set of $\mathbb{T}'$ into a family of paths, with at most $|\mathbf{I}(\mathbb{T})| - 1 \leq 2\rho - 1$ paths of positive length; let $\mathcal{P}(\mathbb{T}')$ be the family of all these paths. The *rounded length of* $\mathbb{T}'$, denoted $\mathtt{rnd}(\mathbb{T}')$, equals $\sum_{P \in \mathcal{P}(\mathbb{T}')} \mathtt{rnd}(w(P))$. Observe that

$$w(\mathbb{T}') \leq \lambda \mathtt{rnd}(\mathbb{T}') \leq w(\mathbb{T}') + (2\rho - 1)\lambda. \tag{6}$$

We remark that this bound on $\mathtt{rnd}(\mathbb{T}')$ applies in particular to a $\mathbb{T}'$ that is a path between some branching vertex of $\mathbb{T}$ and a leaf of $\mathbb{T}$.

We now adjust the definition of niceness to the rounded distances. An embedded tree $(\mathbb{T}, \pi)$ is $\mathtt{rnd}$-τ'-*nice* if, for any two consecutive leaves $\widehat{l_a}, \widehat{l_b}$ in $\mathbb{T}$ the following holds. Let $\pi(\widehat{l_a}) = \widehat{a}$ and $\pi(\widehat{l_b}) = \widehat{b}$ and let l_a, l_b be the parents of $\widehat{l_a}, \widehat{l_b}$ in $\mathbb{T}$, respectively; note that $\pi(l_a) = a$ and $\pi(l_b) = b$, possibly $a = b$. Let w be the lowest common ancestor of $\widehat{l_a}$ and $\widehat{l_b}$ in $\mathbb{T}$. Then the requirement for $\mathtt{rnd}$-τ'-niceness is that

$$w(\partial B[a, b]) + \lambda \mathtt{rnd}(\mathbb{T}[w, l_a]) + \lambda \mathtt{rnd}(\mathbb{T}[w, l_b]) \leq (1 - \tau') w(\partial B). \tag{7}$$

An embedded tree is *fully* $\mathtt{rnd}$-τ'-*nice* if additionally (7) holds for $\widehat{l_a}$ being the last leaf of $\mathbb{T}$, $\widehat{l_b}$ being the first leaf of $\mathbb{T}$, $w = r(\mathbb{T})$ and $\partial B[a, b]$ replaced by $\partial^{\uparrow} B[a, b]$. Observe the following.

Claim 9.10. *If an embedded tree is (fully)* $\mathtt{rnd}$-τ'*-nice, then it is also (fully)* τ'*-nice. If an embedded tree with at most 7 leaves is (fully)* τ'*-nice and* $\tau' > \tau/4$*, then it is also (fully)* $\mathtt{rnd}$-$(\tau' - \tau/4)$*-nice.*

Proof. The claim follows by applying inequality (6) to $\mathtt{rnd}(\mathbb{T}[w, l_a])$ and $\mathtt{rnd}(\mathbb{T}[w, l_b])$ in condition (7). ⌟

By Claims 9.9 and 9.10, we may restrict ourselves to searching for a fully $\mathtt{rnd}$-$(\tau/2)$-nice embedded tree: in each of these claims we lose only $\tau/4$ on the niceness of the tree.

We are now ready to describe the main table for the dynamic-programming algorithm. Define L to be the largest integer such that $\lambda L \leq (1 + \tau) w(\partial B)$; observe that $L = \mathcal{O}(\tau^{-1})$. For each $v \in V(B)$, indices $1 \leq i_a \leq i_b \leq r$, and integers $0 \leq k_a, k_b, \ell \leq L$, we define the value $F[v, i_a, i_b, \ell, k_a, k_b]$ to be any embedded tree $(\mathbb{T}, \pi)$ that satisfies the following:

1. $(\mathbb{T}, \pi)$ is $\mathtt{rnd}$-$(\tau/2)$-nice;

2. $\pi(r(\mathbb{T})) = v$;

3. $\mathbb{T}$ has $i_b - i_a + 1$ leaves, mapped by π onto $\widehat{a_{i_a}}, \widehat{a_{i_a+1}}, \ldots, \widehat{a_{i_b}}$ in this order;

4. $\mathbb{T}$ has rounded length at most ℓ;

5. if $\widehat{l_a}$ is the first leaf of $\mathbb{T}$ and $\widehat{l_b}$ is the last leaf, then $\mathtt{rnd}(\mathbb{T}[\widehat{l_a}, r(\mathbb{T})]) \leq k_a$ and $\mathtt{rnd}(\mathbb{T}[\widehat{l_b}, r(\mathbb{T})]) \leq k_b$.

We require that $F[v, i_a, i_b, \ell, k_a, k_b] = \bot$ if no such embedded tree exists.

The next two claims verify that computing all values $F[v, i_a, i_b, \ell, k_a, k_b]$ is sufficient for our needs.

Claim 9.11. *Assume that $F[v, 1, r, L, k_a, k_b] = (\mathbb{T}, \pi) \neq \bot$ for some v, k_a, k_b. Moreover, assume that*

$$w(\partial^{\uparrow} B[a_r, a_1]) + \lambda k_a + \lambda k_b \leq (1 - \tau/2) w(\partial B). \tag{8}$$

Then $(\mathbb{T}, \pi)$ is fully $(\tau/2)$-nice and has length at most $(1+\tau)w(\partial B)$.

Proof. First, observe that $(\mathbb{T}, \pi)$ is $\mathtt{rnd}$-$(\tau/2)$-nice by the properties of the cell $F[v, 1, r, L, k_a, k_b]$. Moreover, we have that the first leaf of $\mathbb{T}$ is mapped onto $\widehat{a_1}$ and the last leaf is mapped onto $\widehat{a_r}$. Hence, inequality (8) implies that $(\mathbb{T}, \pi)$ is fully $\mathtt{rnd}$-$(\tau/2)$-nice. By Claim 9.10, $(\mathbb{T}, \pi)$ is fully $(\tau/2)$-nice. Finally, note that since $\mathbb{T}$ has rounded length at most L, by (6) the length of $\mathbb{T}$ is bounded by $L\lambda \leq (1+\tau)w(\partial B)$. ⌟

Claim 9.12. *Assume that there exists in B a fully $(3\tau/4)$-nice embedded tree $(\mathbb{T}, \pi)$ of length at most $(1 + 7\tau/8)w(\partial B)$, such that the leaves of $\mathbb{T}$ are mapped onto $a_1, a_2, \ldots, a_r$ in this order. Then $F[v, 1, r, L, k_a, k_b] \neq \bot$ for some v and k_a, k_b satisfying (8).*

Proof. First, observe that by (6) we have

$$\lambda \mathtt{rnd}(\mathbb{T}) \leq w(\mathbb{T}) + 13\lambda \leq (1+7\tau/8)w(\partial B) + 13\lambda = (1+7\tau/8)w(\partial B) + \frac{13\tau}{104} w(\partial B) = (1+\tau)w(\partial B).$$

By the definition of L, this means that $\mathtt{rnd}(\mathbb{T}) \leq L$. Let $\widehat{l_a}$ and $\widehat{l_b}$ be the first and the last leaf of $\mathbb{T}$, respectively. Let $k_a = \mathtt{rnd}(\mathbb{T}[\widehat{l_a}, r(\mathbb{T})])$ and $k_b = \mathtt{rnd}(\mathbb{T}[\widehat{l_b}, r(\mathbb{T})])$. Note that $k_a, k_b \leq \mathtt{rnd}(\mathbb{T}) \leq L$. Hence, $(\mathbb{T}, \pi)$ is a valid candidate for $F[v, 1, r, L, k_a, k_b]$ where $v = \pi(r(\mathbb{T}))$. Moreover, by Claim 9.10, $(\mathbb{T}, \pi)$ is fully $\mathtt{rnd}$-$(\tau/2)$-nice and hence inequality (8) is satisfied for k_a and k_b. ⌟

We now describe how to compute the values $F[v, i_a, i_b, \ell, k_a, k_b]$. Initially, we set $F[a_i, i, i, 0, k_a, k_b]$ to be a tree consisting of the edge $\widehat{a_i}a_i$ with the identity mapping, for each $1 \leq i \leq r$ and $0 \leq k_a, k_b \leq L$. Moreover, we set $F[v, i, i, 0, k_a, k_b] = \bot$ for any $v \neq a_i$ and $0 \leq k_a, k_b \leq L$. It is straightforward to verify that these are correct values of the entries $F[v, i_a, i_b, \ell, k_a, k_b]$ for $i_a = i_b$ and $\ell = 0$.

Then, we compute the values $F[v, i_a, i_b, \ell, k_a, k_b]$ in order of increasing values $(i_b - i_a)$ and ℓ. That is, for fixed i_a, i_b, ℓ, k_a, k_b, we want to compute the entries $F[v, i_a, i_b, \ell, k_a, k_b]$ for all $v \in V(B)$ in $\mathcal{O}(\tau^{-4}|B| \log |B|)$ time, assuming that all entries $F[v', i'_a, i'_b, \ell', k'_a, k'_b]$ were already computed whenever $i'_b - i'_a \leq i_b - i_a$, $\ell' \leq \ell$, and at least one of this inequality is strict.

Consider now a cell $F[v, i_a, i_b, \ell, k_a, k_b]$ for $(i_b - i_a) + \ell > 0$. If $F[v, i_a, i_b, \ell', k'_a, k'_b] \neq \bot$ for some $\ell' \leq \ell$, $k'_a \leq k_a$, $k'_b \leq k_b$ and $(\ell, k_a, k_b) \neq (\ell', k'_a, k'_b)$, then we may copy the value of $F[v, i_a, i_b, \ell', k'_a, k'_b]$ and conclude. Hence, assume otherwise.

Consider an embedded tree $(\mathbb{T}, \pi)$ that satisfies all requirements for the cell $F[v, i_a, i_b, \ell, k_a, k_b]$. There are two cases, depending on the degree of $r(\mathbb{T})$.

If $r(\mathbb{T})$ has at least two children in $\mathbb{T}$, let $\mathbb{T}_1$ be the subtree of $\mathbb{T}$ rooted at the first child of $\mathbb{T}$ (together with the edge towards the root $r(\mathbb{T})$) and let $\mathbb{T}_2 = \mathbb{T} \setminus \mathbb{T}_1$. Denote $\pi_j = \pi|_{\mathbb{T}_j}$ for $j = 1, 2$. Let i be such that $\mathbb{T}_1$ has $i - i_a$ leaves, that is, the last leaf of $\mathbb{T}_1$, denoted $\widehat{l_c}$, is mapped onto $\widehat{a_{i-1}}$ and the first leaf of $\mathbb{T}_2$, denoted $\widehat{l_d}$, is mapped onto $\widehat{a_i}$. Observe that $i_a < i \leq i_c$. Denote $\ell_1 = \mathtt{rnd}(\mathbb{T}_1)$, $\ell_2 = \mathtt{rnd}(\mathbb{T}_2)$, $k_c = \mathtt{rnd}(\mathbb{T}_1[r(\mathbb{T}), \widehat{l_c}])$, and $k_d = \mathtt{rnd}(\mathbb{T}_2[r(\mathbb{T}), \widehat{l_d})$. Observe that $(\mathbb{T}_1, \pi_1)$ is a feasible entry for $F[v, i_a, i-1, \ell_1, k_a, k_c]$ and $(\mathbb{T}_2, \pi_2)$ is a feasible entry for

$F[v, i, i_b, \ell_2, k_d, k_b]$. Moreover, $\ell_1, \ell_2 \leq \ell$, $\ell_1 + \ell_2 = \ell$ and, since $(\mathbb{T}, \pi)$ is $\mathtt{rnd}$-$(\tau/2)$-nice we have that

$$w(\partial B[a_{i-1}, a_i]) + \lambda k_c + \lambda k_d \leq (1 - \tau/2) w(\partial B). \tag{9}$$

In the other direction, assume that for some choice of $0 \leq \ell_1, \ell_2 \leq \ell$ with $\ell_1 + \ell_2 = \ell$, $i_a < i \leq i_b$ and $0 \leq k_c, k_d \leq L$ satisfying (9) we have $F[v, i_a, i-1, \ell_1, k_a, k_c] = (\mathbb{T}_1, \pi_1) \neq \bot$ and $F[v, i, i_b, \ell_2, k_d, k_b] = (\mathbb{T}_2, \pi_2) \neq \bot$. Define $\mathbb{T}$ to be $\mathbb{T}_1 \cup \mathbb{T}_2$ with identified roots of $\mathbb{T}_1$ and $\mathbb{T}_2$, and $\pi = \pi_1 \cup \pi_2$. It is straightforward to verify that $(\mathbb{T}, \pi)$ is a feasible entry for $F[v, i_a, i_b, \ell, k_a, k_b]$. Here observe that (9) ensures that condition (7) is satisfied for leaves mapped to $\widehat{a_{i-1}}$ and $\widehat{a_i}$. Moreover, in the dynamic programming the values $F[v, i_a, i-1, \ell_1, k_a, k_c]$ and $F[v, i, i_b, \ell_2, k_d, k_b]$ are already computed when we consider the cell $F[v, i_a, i_b, \ell, k_a, k_b]$, since $i - 1 - i_a < i_b - i_a$, $i_b - i < i_b - i_a$ and $\ell_1, \ell_2 \leq \ell$. Hence, we look for a feasible candidate for $F[v, i_a, i_b, \ell, k_a, k_b]$ among all values $\ell_1, \ell_2, i, k_c, k_d$ as above and merge $(\mathbb{T}_1, \pi_1)$ with $(\mathbb{T}_2, \pi_2)$ whenever possible. By the argumentation so far, whenever there exists a feasible candidate for $F[v, i_a, i_b, \ell, k_a, k_b]$ with root of degree at least two, we find at least one such candidate.

In the remaining case, $r(\mathbb{T})$ has exactly one child. Observe that $\mathbb{T}$ has more than one edge, as otherwise $i_a = i_b$, $v = a_{i_a}$ and $(\mathbb{T}, \pi)$ is a feasible candidate for $F[v, i_a, i_b, 0, 0, 0]$, and we would have found $(\mathbb{T}, \pi)$ in the first step. Hence, $\mathbf{I}(\mathbb{T})$ contains at least two vertices. Let x be the vertex of $\mathbf{I}(\mathbb{T}) \setminus r(\mathbb{T})$ that is closest to $r(\mathbb{T})$. Denote $u = \pi(x)$ and $k = \mathtt{rnd}(w(\mathbb{T}[r(\mathbb{T}), x]))$; note that $k > 0$. Define $\mathbb{T}'$ to be the tree $\mathbb{T} \setminus \mathbb{T}[r(\mathbb{T}), x]$, rooted at x, and $\pi' = \pi|_{\mathbb{T}'}$. Observe that $(\mathbb{T}', \pi')$ is an embedded tree and it is a feasible candidate for $F[u, i_a, i_b, \ell - k, k_a - k, k_b - k]$.

In the other direction, assume that $F[u, i_a, i_b, \ell - k, k_a - k, k_b - k] = (\mathbb{T}', \pi')$ for some $u \in V(B)$, where $k \geq \mathtt{rnd}(\mathrm{dist}_B(u, v))$. To obtain an embedded tree $(\mathbb{T}, \pi)$, extend $(\mathbb{T}', \pi')$ with a copy of a shortest path between u and v in B, mapped by π to its original, connecting $r(\mathbb{T}')$ with a new root $r(\mathbb{T})$ (mapped by π to v). It is straightforward to verify that $(\mathbb{T}, \pi)$ is a feasible candidate for $F[v, i_a, i_b, \ell, k_a, k_b]$. We remark here that the rounded length of $\mathbb{T}$ may be strictly smaller than $\mathtt{rnd}(\mathbb{T}') + \mathtt{rnd}(\mathrm{dist}_B(u, v))$ in the case when $r(\mathbb{T}')$ has degree one.

Hence, to verify whether there exists a feasible candidate for $F[v, i_a, i_b, \ell, k_a, k_b]$, we need to inspect all entries $F[u, i_a, i_b, \ell - k, k_a - k, k_b - k]$ where $u \in V(B) \setminus \{v\}$ and $k \geq \mathtt{rnd}(\mathrm{dist}_B(u, v))$. However, a naive implementation would take time quadratic in $|B|$. We now show how to check all pairs (v, u) using at most L runs of Dijkstra's shortest-path algorithm in B, which yields a $\mathcal{O}(L|B| \log |B|)$-time algorithm. Iterate through all integers k such that $1 \leq k \leq \min(\ell, k_a, k_b) \leq L$. Define U to be the set of these vertices u for which $F[u, i_a, i_b, \ell - k, k_a - k, k_b - k] \neq \bot$. By a single run of Dijkstra's algorithm in B starting from U, we may compute $\mathrm{dist}_B(v, U)$ for every $v \in V(B)$. Moreover, for each $v \in V(B)$ we can compute the closest vertex $u(v) \in U$ and a shortest path between v and $u(v)$. Then we inspect all $v \in V(B)$ and whenever $\mathtt{rnd}(\mathrm{dist}_B(v, U)) \leq k$, we may use the entry $F[u(v), i_a, i_b, \ell - k, k_a - k, k_b - k]$ to find a feasible candidate for $F[v, i_a, i_b, \ell, k_a, k_b]$.

We remark here that we do not need to explicitly keep the embedded trees as values of $F[v, i_a, i_b, \ell, k_a, k_b]$. It suffices to keep only a boolean that signals whether a feasible candidate has been found and, if this is the case, how it was obtained. Then, the actual tree for a fixed cell $F[v, i_a, i_b, \ell, k_a, k_b]$ can be computed in $\mathcal{O}(|B|)$ time: we need to reproduce at most 13 shortest paths in the tree, each of which can be computed in linear time [45].

We now analyze the running time. There is an $\mathcal{O}(\tau^{-7})$ overhead from guessing r and the sequence $a_1, a_2, \ldots, a_r$. In the dynamic-programming algorithm, in each step we need to keep track of at most 7 integer variables ranging from 0 to L (namely, $\ell, k_a, k_b, \ell_1, \ell_2, k_c, k_d$). Recall that $r \leq 7$. Hence, we obtain a running time of $\mathcal{O}(\tau^{-14}|B| \log |B|)$. □

We may now conclude the proof of Theorem 9.2. By Lemma 9.7, if a short τ-nice tree exists in B, then there exists a fully τ-nice embedded tree with at most 7 leaves and not larger length.

Using Lemma 9.8 we look for such a tree; if it indeed exists in B, we obtain a fully $(\tau/2)$-nice embedded tree of length at most $(1+\tau)w(\partial B)$. In this case, we apply Lemma 9.4 to obtain the desired family of bricks. If the algorithm of Lemma 9.8 does not find any embedded tree, Lemma 9.7 allows us to conclude that no short τ-nice tree exists in B.

10 Weighted variant

We now focus on the weighted variant, and prove Theorem 1.7. As described in the outline of the proof of Theorem 1.7 (see Section 2.4), we first develop a base case: we prove that Theorem 1.7 holds if $\mathcal{S}$ consists of a single pair and only a multiplicative error in the weight of the forest F_H is allowed. A precise statement and the proof of this result is given in Section 10.1. Then, in Section 10.2, we present the θ-variant of Theorem 1.7, where $\mathcal{S}$ contains at most θ terminal pairs and $w(H)$ depends polynomially on ϵ^{-1} and θ. Finally, in Section 10.3, we derive Theorem 1.7.

10.1 Single-pair case

The goal of this section is to prove the following single-pair variant of Theorem 1.7:

Theorem 10.1. *Let $\varepsilon > 0$ be a fixed accuracy parameter, and let B be an edge-weighted brick. Then one can find in $\mathcal{O}(\varepsilon^{-2}|B|\log|B|)$ time a graph $H \subseteq B$ such that*

1. *$\partial B \subseteq H$,*
2. *$w(H) = \mathcal{O}(\varepsilon^{-5}w(\partial B))$, and*
3. *for any pair of vertices $s, t \in V(\partial B)$ there exists a path connecting s and t in H of weight at most $(1+\varepsilon)\mathrm{dist}_B(s,t)$.*

We prove Theorem 10.1 in two steps. First, we observe that it suffices to look only at special types of bricks, called *strips*, introduced by Klein [51]. Intuitively, each strip looks like a long eye-shaped brick, with its perimeter being a union of two (almost) shortest paths. Second, we show that in a strip it suffices to take as H only a small number of shortest paths, somewhat resembling vertical ‘columns’ in an eye-shaped strip.

10.1.1 Strips

Let $\varepsilon > 0$ be fixed and, without loss of generality, assume $\varepsilon < \frac{1}{2}$. We need the following notions from [51, 12] (phrased in our terminology).

Definition 10.2 (ε-short path, [12]). *A path P in a graph G is* ε-short in G *if for every two vertices $x, y \in V(P)$ we have $\mathrm{dist}_P(x,y) \leq (1+\varepsilon)\mathrm{dist}_G(x,y)$.*

Definition 10.3 (strip, [51, 12]). *A brick B is called a* strip *if ∂B can be decomposed into two nonempty paths $\mathbf{N}$ and $\mathbf{S}$ such that in B: (a) $\mathbf{S}$ is a shortest path between its endpoints, and (b) every proper subpath of $\mathbf{N}$ is an ε-short path.*

The paths $\mathbf{N}$ and $\mathbf{S}$ are called *north* and *south* boundary of the strip B. If one travels ∂B in the counter-clockwise direction, the first vertex of $\mathbf{S}$ is called the *leftmost* vertex of $\mathbf{S}$ and the last vertex is the *rightmost* vertex of $\mathbf{S}$. Note that both these vertices are also endpoints of $\mathbf{N}$.

Lemma 10.4 ([51, 12]). *There exists an algorithm that, given a brick B, in $\mathcal{O}(|B|\log|B|)$ time finds a $(2\varepsilon^{-1}+3)$-short brick partition $\mathcal{B} = \{B_1, \ldots, B_p\}$ of B, such that B_i is a strip for every $1 \leq i \leq p$. Moreover, within the same running time, the algorithm can also provide the partition of ∂B_i into paths $\mathbf{N}$ and $\mathbf{S}$ for every $1 \leq i \leq p$.*

Using Lemma 10.4, we can show that it suffices to prove the following theorem:

Theorem 10.5. *Let $\varepsilon > 0$ be a fixed accuracy parameter, and let B be an edge-weighted strip with given north and south boundary. Then one can find in $\mathcal{O}(\varepsilon^{-2}|B|\log|B|)$ time a graph $H \subseteq B$ such that*

1. *$\partial B \subseteq H$,*
2. *$w(H) = \mathcal{O}(\varepsilon^{-4}w(\partial B))$, and*
3. *for any pair of vertices $s, t \in V(\partial B)$ there exists a path connecting s and t in H of length at most $(1+\varepsilon)\mathrm{dist}_B(s,t)$.*

Indeed, using Lemma 10.4 and Theorem 10.5, we can prove Theorem 10.1 as follows:

Proof of Theorem 10.1. Apply Lemma 10.4 to the input brick B, and let $\mathcal{B} = \{B_1, \ldots, B_p\}$ be the resulting brick partition, where each B_i is a strip. Apply the algorithm of Theorem 10.5 to each strip B_i, and let H_i denote the resulting subgraph of B_i. We claim that $H := \bigcup_{i=1}^{p} H_i$ satisfies all requirements of Theorem 10.1. Since $\mathcal{B}$ is a brick partition of B, $\partial B \subseteq H$. Moreover, H can be computed in total $\mathcal{O}(\varepsilon^{-2}|B|\log|B|)$ time due to the time bounds of Lemma 10.4 and Theorem 10.5, and the fact that each edge of B belongs to at most two strips B_i (since $\mathcal{B}$ is a brick partition of B).

Consider any pair $s, t \in V(\partial B)$ and let Q be a shortest path between s and t in the brick B. Let Q' be any maximal subpath of Q that consists only of edges strictly enclosed by ∂B_i for some $1 \le i \le p$, and let s', t' be the endpoints of Q'. By the properties of H_i, there exists a path $P' \subseteq H_i$ of length at most $(1+\varepsilon)w(P')$ with endpoints s' and t'. Moreover, observe that all such subpaths Q' of Q are pairwise edge-disjoint. Hence, we may replace each such subpath Q' with the path P' defined as above, obtaining a walk P connecting s and t of length at most $(1+\varepsilon)w(Q)$. As $\partial B_i \subseteq H$ for each $1 \le i \le p$ and each constructed path P' is contained in some graph H_i, we have $P \subseteq H$. Finally, observe that since $\mathcal{B}$ is $(2\varepsilon^{-1}+3)$-short,

$$w(\bigcup_{i=1}^{p} H_i) \le \sum_{i=1}^{p} w(H_i) \le \sum_{i=1}^{p} \mathcal{O}(\varepsilon^{-4}w(\partial B_i)) = \mathcal{O}(\varepsilon^{-5}w(\partial B)).$$

The theorem follows. □

10.1.2 Columns

In the remainder of this section, we aim to prove Theorem 10.5. Let B denote the strip, and let $\mathbf{N}$ and $\mathbf{S}$ denote the partition of ∂B as per definition. Let $\mathfrak{l}$ and $\mathfrak{r}$ be the leftmost and rightmost vertex of $\mathbf{S}$, respectively. Recall that $\mathfrak{l}$ and $\mathfrak{r}$ are endpoints of both $\mathbf{N}$ and $\mathbf{S}$.

We now identify *columns* in the strip B in a similar, but slightly different, way as in [51, 12]. Define $x_0 := \mathfrak{l}$ and traverse $\mathbf{N}$ starting from the endpoint x_0. For $i = 1, 2, \ldots$, find the first vertex x_i on $\mathbf{N}$ such that $\mathrm{dist}_{\mathbf{N}}(x_{i-1}, x_i) > \varepsilon\mathrm{dist}_B(x_{i-1}, \mathbf{S})$. Define column C_i to be some arbitrarily chosen shortest path between x_i and $\mathbf{S}$ in B and let y_i be the other endpoint of C_i. Let x_q be the last vertex defined in this way. Observe that $x_q = \mathfrak{r}$, as otherwise $\mathrm{dist}_{\mathbf{N}}(x_q, \mathfrak{r}) \ge \mathrm{dist}_B(x_q, \mathfrak{r}) \ge \mathrm{dist}_B(x_q, \mathbf{S})$ and, since $\varepsilon < \frac{1}{2}$, the vertex $\mathfrak{r}$ is a feasible candidate for x_{q+1}.

Similarly as in [51], we can bound the total length of columns and find them efficiently.

Lemma 10.6. *The sum of the length of the columns is bounded by $\varepsilon^{-1}w(\mathbf{N}) \le \varepsilon^{-1}w(\partial B)$.*

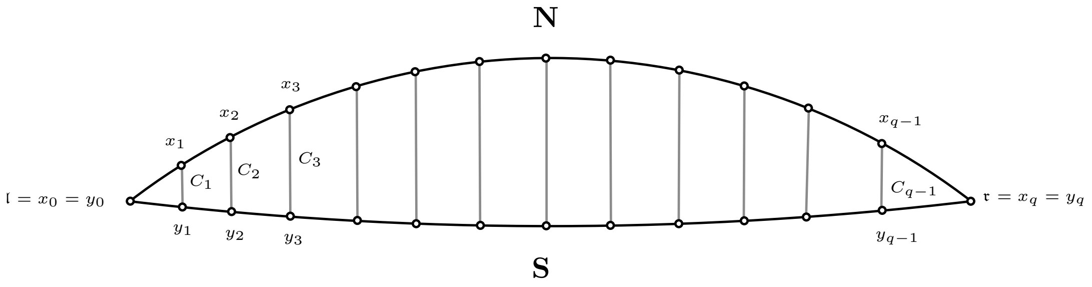


Figure 9: Construction of columns in a strip.

Proof. As $x_q = \mathfrak{r}$, we have $w(C_q) = 0$. For the other columns, by construction and the definition of $\mathbf{N}$:

$$\sum_{i=0}^{q-1} w(C_i) = \sum_{i=0}^{q-1} \mathrm{dist}_B(x_i, \mathbf{S}) < \varepsilon^{-1}\mathrm{dist}_{\mathbf{N}}(x_i, x_{i+1}) = \varepsilon^{-1} w(\mathbf{N}).$$

□

Lemma 10.7. *One can identify the vertices $x_0, x_1, \ldots, x_q$, distances* $\mathrm{dist}_B(x_i, \mathbf{S})$ *for $0 \le i \le q$, vertices $y_0, y_1, \ldots, y_q$, and the edge set $\bigcup_{i=0}^{q} C_i$ for some choice of columns C_i, in $\mathcal{O}(|B|)$ time.*

Proof. Create a new super-terminal t in the outer face of B, connect it with zero-length edges to all vertices of $V(\mathbf{S})$, and find a shortest-path tree T rooted at t in the resulting planar graph. This takes linear time [45]. Using the tree, one can identify for each $v \in V(B)$ both the distance $\mathrm{dist}_B(v, \mathbf{S})$ and some closest vertex $y(v) \in V(\mathbf{S})$. With this information, it is straightforward to identify the vertices x_i, y_i and values $\mathrm{dist}_B(x_i, \mathbf{S})$, where the column C_i is taken as the subpath in T between the vertex x_i and its ancestor $y_i := y(x_i) \in V(\mathbf{S})$ on the path to the root t. Moreover, by traversing the tree T in a bottom-up fashion, one can in linear time mark all the edges of $\bigcup_{i=0}^{q} C_i$ for this particular choice of columns C_i. □

We remark that the columns C_i computed by the algorithm of Lemma 10.7 are non-crossing in the following sense: whenever C_i and C_j meet at some point, they continue together to a common endpoint $y_i = y_j$. However, we will not use this property in the sequel.

10.1.3 Pegs

The idea of the construction of the subgraph H of B is as follows: either the shortest path between the given terminals s and t goes along one column C_i, or it is more-or-less horizontal. In the first case, we hope to handle it by adding to H a few shortest paths that are close to C_i. In the second case, some column C_i should have negligible length compared to the length of such a path, and thus the columns C_i together with ∂B should suffice to handle such a pair (s, t).

We now establish how to handle terminal pairs (s, t) where the shortest path between s and t in B is in some sense close to some column C_i. Fix one index i, $0 \le i \le q$, and define $z_{i,0} := y_i$. Traverse $\mathbf{S}$ to the right (in the direction of $\mathfrak{r}$), starting from $z_{i,0}$. For each $j = 1, 2, \ldots$ define $z_{i,j}$ to be the first vertex satisfying $\mathrm{dist}_{\mathbf{S}}(z_{i,j-1}, z_{i,j}) > \varepsilon w(C_i)$. Symmetrically, traverse $\mathbf{S}$ to the left (in the direction of $\mathfrak{l}$), starting from $z_{i,0}$. For each $j = 1, 2, \ldots$, define $z_{i,-j}$ to be the first vertex satisfying $\mathrm{dist}_{\mathbf{S}}(z_{i,-(j-1)}, z_{i,-j}) > \varepsilon w(C_i)$. The vertices $z_{i,j}$ are called *pegs* for the column C_i. Define $r_{\rightarrow}(i)$ to be the last index j for which $\mathrm{dist}_{\mathbf{S}}(y_i, z_{i,j}) \le \varepsilon^{-1} w(C_i)$, and $r_{\leftarrow}(i)$ to be the last index j for which $\mathrm{dist}_{\mathbf{S}}(y_i, z_{i,-j}) \le \varepsilon^{-1} w(C_i)$. Observe that

$$r_{\rightarrow}(i), r_{\leftarrow}(i) \le \varepsilon^{-2}. \tag{10}$$

For every $-r_{\leftarrow}(i) \leq j \leq r_{\rightarrow}(i)$ define the *cord* $P_{i,j}$ to be some arbitrarily chosen shortest path in B between x_i and $z_{i,j}$. In what follows we show that

$$H := \partial B \cup \bigcup_{i=0}^{q} \bigcup_{j=-r_{\leftarrow}(i)}^{r_{\rightarrow}(i)} P_{i,j} \tag{11}$$

satisfies all the desired conditions for the subgraph H and, moreover, it can be computed efficiently for some choice of cords $P_{i,j}$.

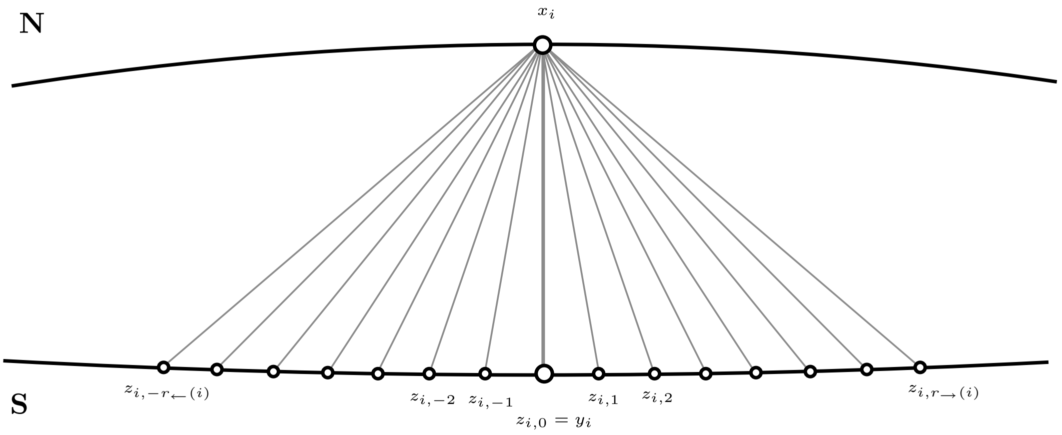


Figure 10: Construction of pegs $z_{i,j}$ and cords $P_{i,j}$ (in grey) for one column C_i.

10.1.4 Correctness

We first show that the graph H defined in (11) preserves approximate shortest paths between pairs of vertices on ∂B.

Lemma 10.8. *For any $s, t \in V(\partial B)$, it holds that* $\mathrm{dist}_H(s,t) \leq (1 + 12\varepsilon)\mathrm{dist}_B(s,t)$.

Proof. If both s and t belong to $V(\mathbf{S})$ or both s and t belong to $V(\mathbf{N})$, then the claim is straightforward, as $\partial B \subseteq H$, $\mathbf{S}$ is a shortest path in B, and any proper subpath of $\mathbf{N}$ is ε-short in B. Hence, we may restrict ourselves to the case $s \in V(\mathbf{N})$, $t \in V(\mathbf{S})$.

Let $0 \leq i < q$ be an index such that s lies on the subpath of $\mathbf{N}$ between x_i and x_{i+1} (possibly $s = x_i$, but $s \neq x_{i+1}$). By the choice of x_{i+1},

$$\mathrm{dist}_B(s, x_i) \leq \mathrm{dist}_{\mathbf{N}}(s, x_i) \leq \varepsilon\, w(C_i) = \varepsilon\, \mathrm{dist}_B(x_i, \mathbf{S}). \tag{12}$$

As $t \in V(\mathbf{S})$, it follows from (12) that

$$\mathrm{dist}_B(s,t) \geq \mathrm{dist}_B(s, \mathbf{S}) \geq \mathrm{dist}_B(x_i, \mathbf{S}) - \mathrm{dist}_B(s, x_i) \geq (1 - \varepsilon) w(C_i). \tag{13}$$

Without loss of generality, assume that t lies on $\mathbf{S}$ between $y_i = z_{i,0}$ and $\mathfrak{r}$ (possibly $t = y_i$); the second case will be symmetrical. We now consider two cases, depending on $\mathrm{dist}_{\mathbf{S}}(y_i, t)$. If $\mathrm{dist}_{\mathbf{S}}(y_i, t) \leq \varepsilon^{-1} w(C_i)$, then let j be the last index such that $z_{i,j}$ lies on the path $\mathbf{S}$ between y_i and t (possibly $t = z_{i,j}$). By the definition of the sequence $z_{i,j}$, we have $\mathrm{dist}_{\mathbf{S}}(z_{i,j}, t) \leq \varepsilon w(C_i)$. By the definition of $r_{\rightarrow}(i)$, we have $j \leq r_{\rightarrow}(i)$. Note that the cord $P_{i,j}$ is contained in H. Consider the following path between s and t in H: we first go along $\mathbf{N}$ to x_i, then traverse $P_{i,j}$ to $z_{i,j}$, and then continue along $\mathbf{S}$ to t. We obtain:

$$\mathrm{dist}_H(s,t) \leq \mathrm{dist}_{\mathbf{N}}(s, x_i) + w(P_{i,j}) + \mathrm{dist}_{\mathbf{S}}(z_{i,j}, t) \leq 2\varepsilon w(C_i) + \mathrm{dist}_B(x_i, z_{i,j}). \tag{14}$$

However,

$$\mathrm{dist}_B(x_i, z_{i,j}) \leq \mathrm{dist}_B(x_i, s) + \mathrm{dist}_B(s,t) + \mathrm{dist}_B(t, z_{i,j}) \leq 2\varepsilon w(C_i) + \mathrm{dist}_B(s,t). \tag{15}$$

By combining (13), (14) and (15) we obtain

$$\mathrm{dist}_H(s,t) \leq 4\varepsilon w(C_i) + \mathrm{dist}_B(s,t) \leq \left(1 + \frac{4\varepsilon}{1-\varepsilon}\right)\mathrm{dist}_B(s,t) < (1+8\varepsilon)\mathrm{dist}_B(s,t),$$

where the last inequality follows from the assumption that $\varepsilon < \frac{1}{2}$.

In the remaining case, $\mathrm{dist}_{\mathbf{S}}(y_i, t) > \varepsilon^{-1} w(C_i)$ and, consequently,

$$\begin{aligned} \mathrm{dist}_B(s,t) &\geq \mathrm{dist}_B(y_i,t) - \mathrm{dist}_B(x_i,y_i) - \mathrm{dist}_B(s,x_i) \\ &\geq \mathrm{dist}_{\mathbf{S}}(y_i,t) - (1+\varepsilon)w(C_i) \\ &> (\varepsilon^{-1} - 1 - \varepsilon)w(C_i). \end{aligned} \tag{16}$$

Moreover,

$$\begin{aligned} \mathrm{dist}_{\mathbf{S}}(y_i,t) &= \mathrm{dist}_B(y_i,t) \\ &\leq \mathrm{dist}_B(s,t) + \mathrm{dist}_B(s,x_i) + \mathrm{dist}_B(x_i,y_i) \\ &\leq \mathrm{dist}_B(s,t) + (1+\varepsilon)w(C_i). \end{aligned} \tag{17}$$

Consider the following path connecting s and t in H: we first traverse $\mathbf{N}$ from s to x_i, then we go along the cord $P_{i,0}$, and then we traverse $\mathbf{S}$ from y_i to t. Since this path exists in H and by (16) and (17), we obtain that:

$$\begin{aligned} \mathrm{dist}_H(s,t) &\leq \mathrm{dist}_{\mathbf{N}}(s,x_i) + w(P_{i,0}) + \mathrm{dist}_{\mathbf{S}}(y_i,t) \\ &\leq (1+\varepsilon)w(C_i) + \mathrm{dist}_B(s,t) + (1+\varepsilon)w(C_i) \\ &\leq \mathrm{dist}_B(s,t) + (2+2\varepsilon)w(C_i) \\ &< \mathrm{dist}_B(s,t) + \frac{2+2\varepsilon}{\varepsilon^{-1}-1-\varepsilon}\mathrm{dist}_B(s,t) \\ &< (1+12\varepsilon)\mathrm{dist}_B(s,t), \end{aligned}$$

where the last inequality follows from the assumption that $\varepsilon < \frac{1}{2}$. The lemma follows. □

10.1.5 Bound on the length of H

We now bound the total length of the graph H.

Lemma 10.9. *The graph H defined in* (11) *has total length $\mathcal{O}(\varepsilon^{-4} w(\partial B))$.*

Proof. Observe that for any vertex x_i and any peg $z_{i,j}$ with $-r_{\leftarrow}(i) \leq j \leq r_{\rightarrow}(i)$ we have

$$w(P_{i,j}) = \mathrm{dist}_B(x_i, z_{i,j}) \leq \mathrm{dist}_B(x_i, y_i) + \mathrm{dist}_B(y_i, z_{i,j}) \leq (1+\varepsilon^{-1})w(C_i). \tag{18}$$

Hence,

$$\begin{aligned} w(H) &\leq w(\partial B) + \sum_{i=0}^{q} \sum_{j=-r_{\leftarrow}(i)}^{r_{\rightarrow}(i)} w(P_{i,j}) && \text{by definition of } H \\ &\leq w(\partial B) + \sum_{i=0}^{q} \left(1 + r_{\leftarrow}(i) + r_{\rightarrow}(i)\right)(1+\varepsilon^{-1})w(C_i) && \text{by (18)} \\ &\leq w(\partial B) + \sum_{i=0}^{q} (1+\varepsilon^{-1})(1+2\varepsilon^{-2})w(C_i) && \text{by (10)} \\ &\leq w(\partial B) + (1+\varepsilon^{-1})(1+2\varepsilon^{-2})\varepsilon^{-1}w(\partial B) && \text{by Lemma 10.6} \\ &= \mathcal{O}(\varepsilon^{-4}w(\partial B)) && \text{as } \varepsilon < \frac{1}{2}. \end{aligned}$$

□

10.1.6 Efficient computation of H

To finish the proof of Theorem 10.5, we need to show how to compute the graph H in $\mathcal{O}(\varepsilon^{-2}|B|\log|B|)$ time. Observe that a naive implementation uses $\mathcal{O}(|B|\log|B|)$ time for each vertex x_i, giving a $\mathcal{O}(|B|^2\log|B|)$ time bound. This is because we do not control q, the number of vertices x_i. We improve the running time using the multiple-source shortest path algorithm of Klein [50].

Lemma 10.10. *The graph H defined in* (11) *can be computed in $\mathcal{O}(\varepsilon^{-2}|B|\log|B|)$ time.*

Proof. We first recall the multiple-source shortest path algorithm of Klein [50]. Given a brick B with $V(\partial B) = \{v_1, v_2, \ldots, v_s\}$ and vertices v_i appearing on ∂B clockwise in this order, the algorithm computes in $\mathcal{O}(|B|\log|B|)$ time a shortest-path tree $T(v_i)$ for each vertex v_i. However, as the total size of all these shortest-path trees is $\Omega(|B|^2)$, they are computed implicitly. More precisely, in [50] it is shown that in the sequence of rooted trees $T(v_1), T(v_2), \ldots, T(v_s)$ each edge of B is added to a tree or removed from a tree only a constant number of times, and the algorithm of [50] computes in the promised time the sequence of these changes.

We will compute the graph H using an auxiliary variable graph $H^\star$, initially setting $H^\star := \partial B$. At the end of the algorithm, we will have $H = H^\star$. The algorithm traverses the path $\mathbf{N}$ from $\mathfrak{l}$ to $\mathfrak{r}$, running the algorithm of [50] to maintain a shortest-path tree $T(v)$ from the currently considered vertex $v \in V(\mathbf{N})$, and computing the union of all the paths $P_{i,j}$ along the way. We will maintain the invariant that once vertex x_i is processed, the graph $H^\star$ contains ∂B and all the paths $P_{i',j}$ for all $i' \leq i$.

When visiting a vertex $v \in V(\mathbf{N})$, apart from the tree $T(v)$ maintained by the algorithm of [50], we also maintain the graph $T(v) \cap E(H^\star)$. This graph is a forest, so we may maintain it as a dynamic forest in a standard manner, e.g., using ET-trees [44], which allows edge addition and removal in $\mathcal{O}(\log|B|)$ time. Moreover, each tree of $T(v) \cap E(H^\star)$ keeps track of its closest to root vertex in $T(v)$; observe that this information is easy to update upon one edge modification in $T(v)$, since such a modification always affects at most two trees in the forest $T(v) \cap E(H)$.

Assume now that we consider $v = x_i$ for some $0 \leq i \leq q$. Having updated $T(x_i)$ and $T(x_i) \cap E(H^\star)$, we would like to add to $H^\star$ all the edges of $\bigcup_{-r_{\leftarrow}(i) \leq j \leq r_{\rightarrow}(i)} E(P_{i,j})$ that are not yet present in $H^\star$. We first compute all the vertices $z_{i,j}$ for $-r_{\leftarrow}(i) \leq j \leq r_{\rightarrow}(i)$: each $z_{i,j}$ can be found in $\mathcal{O}(\log|B|)$ time using binary search, provided that for each vertex of $\mathbf{S}$ we precomputed its distance from $\mathfrak{l}$ on $\mathbf{S}$. Now, for each j, $-r_{\leftarrow}(i) \leq j \leq r_{\rightarrow}(i)$, we define the cord $P_{i,j}$ as the path between $z_{i,j}$ and x_i in the tree $T(x_i)$.

The procedure of adding the edges of $\bigcup_{-r_{\leftarrow}(i)\leq j\leq r_{\rightarrow}(i)} E(P_{i,j})$ that are not yet present in $H^{\star}$ will use $\mathcal{O}(\log|B|)$ time per each added edge, and thus the total time spent on running all these procedures will be $\mathcal{O}(|B|\log|B|)$. The update of $H^{\star}$ is done as follows. For each j, we traverse the tree $T(x_i)$ from $z_{i,j}$ up to the root x_i, adding to $H^{\star}$ the edges that are missing in $H^{\star}$, and update $T(x_i) \cap E(H^{\star})$ upon each such addition. When traversing the tree, however, we do not consider edges that are already present in $H^{\star}$: using the fact that every component C of $T(x_i) \cap E(H)$ remembers its closest to the root vertex in $T(x_i)$, we may always jump immediately to this vertex, and thus process only edges that are not yet in $H^{\star}$.

Thus, the total work of the algorithm consists of summand $\mathcal{O}(|B|\log|B|)$ for running the algorithm of [50], summand $\mathcal{O}(|B|\log|B|)$ for maintaining the dynamic forest $T(v) \cap E(H^{\star})$ upon modifications of $T(v)$ and of $E(H^{\star})$, and summand $\mathcal{O}(\varepsilon^{-2}|B|\log|B|)$ for computing all the vertices $z_{i,j}$ for all $1 \leq i \leq q$ and $-r_{\leftarrow}(i) \leq j \leq r_{\rightarrow}(i)$. The running time follows. □

By appropriately rescaling the accuracy parameter ε, we complete the proof of Theorem 10.5, and thus of Theorem 10.1.

10.2 Bounded number of terminal pairs

We now take the next step, and prove the θ-variant of Theorem 1.7. To be precise, we show:

Theorem 10.11. *Let $\varepsilon > 0$ be a fixed accuracy parameter, let θ be a positive integer, and let B be an edge-weighted brick. Then one can find in $\mathrm{poly}(\varepsilon^{-1}, \theta)|B|\log|B|$ time a graph $H \subseteq B$ such that*

(i) $\partial B \subseteq H$,

(ii) $w(H) \leq \mathrm{poly}(\varepsilon^{-1}, \theta) w(\partial B)$, and

(iii) for every set $\mathcal{S} \subseteq V(\partial B) \times V(\partial B)$ of size at most θ, there exists a Steiner forest F_H that connects $\mathcal{S}$ in H such that $w(F_H) \leq w(F_B) + \varepsilon w(\partial B)$ for any Steiner forest F_B that connects $\mathcal{S}$ in B.

From a high-level perspective, we proceed similarly as in Section 8. The algorithm has two phases: in the first phase, we recursively use the decomposition tools developed in the previous sections to compute a brick covering $\mathcal{A}$ of B, where each $B' \in \mathcal{A}$ has the following property: either $w(\partial B')$ is small, or for every set $\mathcal{S} \subseteq V(\partial B) \times V(\partial B)$ of size at most θ, there exist an optimal Steiner forest connecting $\mathcal{S}$ that does not contain any vertex of degree larger than 2 that is strictly enclosed by $\partial B'$.

10.2.1 Phase one: decomposing B

We first initialize a family $\mathcal{A} = \emptyset$. During the course of the algorithm all elements of this family will be subbricks of B. Then we call a procedure `partition` on the input brick B. The description of the procedure `partition`, when called on a subbrick B' of B, is as follows.

Call B' tiny if $w(\partial B') \leq \frac{\varepsilon}{\theta} w(\partial B)$, and *large* otherwise. If B' is tiny, then put B' into $\mathcal{A}$. If B' is large, then invoke the algorithm of Theorem 9.2 for the brick B' and parameter $\tau = \frac{1}{36}$. If the algorithm finds a $(3 + 2\tau)$-short $(\tau/2)$-nice brick covering $\mathcal{B}(B')$ of B', then recursively invoke `partition` on all bricks of $\mathcal{B}(B')$.

If the algorithm of Theorem 9.2 finds that no short τ-nice tree exists in B, then invoke the algorithm of Theorem 5.7 for $\tau = \frac{1}{36}$ and $\delta = 2\tau$ to find the core face $f_{\texttt{core}}$, and then invoke the algorithm of Theorem 7.1 for $\tau = \frac{1}{36}$ and the brick B'. Let C be the cycle found by Theorem 7.1. We find a sequence $p_1, p_2, \ldots, p_s$ of pegs on C such that for any $1 \leq i \leq s$ either p_i, p_{i+1} are two consecutive vertices of C or $w(C[p_i, p_{i+1}]) \leq 2\tau w(\partial B')$ (here we assume $p_{s+1} = p_1$). In a greedy manner (as in Section 7), we can find in linear time a sequence of such pegs with

$$s \leq \frac{1}{\tau} \cdot \frac{w(C)}{w(\partial B')} \leq \frac{16}{\tau^3} = \mathcal{O}(1). \tag{19}$$

Then we find, for each peg p_i, a shortest path P_i between p_i and $V(\partial B')$ that does not contain any edge strictly enclosed by C. Let x_i be the second endpoint of P_i. Observe that we may assume that the paths P_i obtained in this manner are non-crossing in the following sense: whenever P_i and P_j meet at some vertex, they continue together towards a common endpoint $x_i = x_j$ on $V(\partial B)$. Indeed, we can find the vertices x_i by removing all edges and vertices that are strictly enclosed by C, adding a super-terminal s_0 in the outer face, and connecting s_0 to the vertices of ∂B using edges of weight zero. The graph we just constructed is planar, and by constructing a shortest-path tree T for s_0 in this graph (which takes linear time [45]), we can find the vertices x_i in linear time. Then the paths P_i are simply the $p_i x_i$-paths in T. By construction, these paths have the required property.

Now consider any i such that $1 \leq i \leq s$ and $C[p_i, p_{i+1}] \neq \partial B'[x_i, x_{i+1}]$. Let W_i denote the closed walk $P_i \cup C[p_i, p_{i+1}] \cup P_{i+1} \cup \partial B'[x_i, x_{i+1}]$ in B'. Let H_i be the graph consisting of all edges of W_i that neighbour the outer face of W_i treated as a planar graph. By definition, each doubly-connected component of H_i is a cycle or a bridge. For each doubly-connected component that is a cycle, we create a brick consisting of all the edges of B that are enclosed by this cycle. Let $\mathcal{B}_i$ be the family of obtained bricks. Observe that $\mathcal{B}_i$ can be computed in linear time for fixed i and a face of B' is enclosed by some brick of $\mathcal{B}_i$ if and only if it is enclosed by W_i. For each $1 \leq i \leq s$, we recursively call `partition` on all bricks of $\mathcal{B}_i$.

Finally, we put a brick B^C consisting of all edges of B enclosed by C into $\mathcal{A}$.

This concludes the description of the procedure `partition`, and hence the description of the first phase of the algorithm. We now analyse the family $\mathcal{A}$ and the running time of the algorithm.

First, we establish some more notation that will be useful in the analysis. For a fixed call `partition`(B'), by $\mathcal{A}(B')$ we denote all bricks that are inserted into $\mathcal{A}$ during this call, and by $\mathcal{A}^{\downarrow}(B')$ we denote all bricks that are inserted into $\mathcal{A}$ in any call in the subtree of the recursion tree rooted at the call `partition`(B'), including $\mathcal{A}(B')$.

In the case when Theorem 7.1 has been invoked, we denote $\mathcal{B}^r(B') = \bigcup_{i=1}^{s} \mathcal{B}_i$ and $\mathcal{B}(B') = \{B^C\} \cup \mathcal{B}^r(B')$. In the case when Theorem 9.2 returned a brick covering $\mathcal{B}(B')$, we denote also $\mathcal{B}^r(B') = \mathcal{B}(B')$. Observe that, regardless of whether Theorem 7.1 has been invoked or not,

- $\mathcal{B}^r(B')$ is the family of subbricks of B' for which a recursive call has been made;
- $\mathcal{B}(B') = \mathcal{A}(B') \cup \mathcal{B}^r(B')$;
- $\mathcal{B}(B')$ is a brick covering of B' with the additional property that $\bigcup_{B^* \in \mathcal{B}(B')} \partial B^*$ is connected.

Using these properties, we analyse the family $\mathcal{A}$.

Lemma 10.12. *$\mathcal{A}$ is a brick covering of B and, moreover, $\bigcup_{B_1 \in \mathcal{A}} \partial B_1$ is connected.*

Proof. By induction on the recursion tree of procedure `partition`, we prove that for any call `partition`(B'), the family $\mathcal{A}^{\downarrow}(B')$ is a brick covering of B' and, moreover, $\bigcup_{B_1 \in \mathcal{A}^{\downarrow}(B')} \partial B_1$ is connected. This is clearly true in the leaves of the recursion tree when $\mathcal{A}^{\downarrow}(B') = \{B'\}$. In an induction step, observe that the fact that $\mathcal{A}^{\downarrow}(B')$ is a brick covering of B' follows from the fact that $\mathcal{B}(B')$ is a brick covering of B' and the induction hypothesis for all elements of $\mathcal{B}^r(B')$. The fact that $\bigcup_{B_1 \in \mathcal{A}^{\downarrow}(B')} \partial B_1$ is connected follows from the fact that $\bigcup_{B^* \in \mathcal{B}(B')} \partial B^*$ is connected, $\mathcal{B}(B') = \mathcal{A}(B') \cup \mathcal{B}^r(B')$ and the induction hypothesis for all elements of $\mathcal{B}^r(B')$. □

Lemma 10.13. *For every set $\mathcal{S} \subseteq V(\partial B) \times V(\partial B)$ there exists a Steiner forest F connecting $\mathcal{S}$ in B of minimum possible length with the following additional property: for every vertex v of degree at least three in F, there exists some $B_1 \in \mathcal{A}$ such that either*

1. *$v \in V(\partial B_1)$, or*
2. *v is strictly enclosed by ∂B_1 and B_1 is tiny.*

Proof. For any call `partition`(B') in the recursion tree, and for any forest F in B', we say that a vertex v is *lame* if (a) the degree of v in F is at least three, and (b) for any $B_1 \in \mathcal{A}^{\downarrow}(B')$ we have $v \notin V(\partial B_1)$, and (c) if v is strictly enclosed by ∂B_1, then B_1 is large. By induction on the recursion tree of the procedure `partition`, we prove that for any call `partition`(B') and any $\mathcal{S} \subseteq V(\partial B') \times V(\partial B')$ there exists a Steiner forest F connecting $\mathcal{S}$ of minimum possible length that does not contain lame vertices. In the leaves of the recursion tree, the statement is clearly true as $\mathcal{A}^{\downarrow}(B') = \{B'\}$ and B' is tiny.

Consider now a call `partition`(B'), and let $\mathcal{S} \subseteq V(\partial B') \times V(\partial B')$. By Theorem 7.1, there exists a Steiner forest F connecting $\mathcal{S}$ in B' of minimum possible length that additionally satisfies the following: if Theorem 7.1 has been invoked to obtain $\mathcal{B}(B')$, then no vertex of degree at least three in F is strictly enclosed by ∂B^C. Pick such F that minimizes the number of lame vertices. We claim that there are in fact no lame vertices; note that such a claim proves the induction step and finishes the proof of the lemma. Assume the contrary, and let v be any lame vertex for F.

As v is not strictly enclosed by ∂B^C in the case when Theorem 7.1 has been invoked, we infer that there exists $B^* \in \mathcal{B}^r(B')$ such that ∂B^* encloses v. As $v \notin V(\partial B_1)$ for any $B_1 \in \mathcal{A}^{\downarrow}(B')$, ∂B^* strictly encloses v. Consider $F_1 := F \cap B^*$ and let $\mathcal{S}_1$ be the set of pairs (x, y) such that $x, y \in V(F_1) \cap V(\partial B^*)$, $x \neq y$, and x, y belong to the same connected component of F_1. By the induction hypothesis, there exists a forest F_2 connecting $\mathcal{S}_1$ in B^* of length at most $w(F_1)$ that does not contain any lame vertices in B^*. Hence, $F' := (F \setminus F_1) \cup F_2$ is a Steiner forest connecting $\mathcal{S}$ in B' of length at most $w(F)$ that contains a strict subset of the set of lame vertices of F, a contradiction to the choice of F. This finishes the induction step, and concludes the proof of the lemma. □

We now move to the analysis of the efficiency of the algorithm. Our goal is to prove upper bounds on the size of $\mathcal{A}$, on the total length of the perimeters of the bricks in $\mathcal{A}$, and on the running time of phase one.

Lemma 10.14. *Let $i \in \{1, \ldots, s\}$ be such that $C[p_i, p_{i+1}] \neq \partial B'[x_i, x_{i+1}]$. Then $\sum_{B_1 \in \mathcal{B}_i} w(\partial B_1) \leq (1 - 2\tau) w(\partial B')$.*

Proof. Consider the walk $Q_i := P_i \cup C[p_i, p_{i+1}] \cup P_{i+1}$ that connects x_i and x_{i+1}. We claim that:

$$w(Q_i) \leq \left(\tfrac{1}{2} - 2\tau\right) w(\partial B'). \tag{20}$$

Indeed, if $w(C[p_i, p_{i+1}]) \leq 2\tau w(\partial B')$, then as each vertex of C is at distance at most $(\frac{1}{4} - 2\tau) \cdot w(\partial B')$ from $V(\partial B')$ by the construction of C and Theorem 7.1, the paths P_i and P_{i+1} have length at most $(\frac{1}{4} - 2\tau) w(\partial B')$, and the claim follows. Otherwise, by the construction of the pegs, $p_i p_{i+1}$ is an edge of C. Now, the claim follows from the fact that each point of C (and in particular every point of the edge $p_i p_{i+1}$) is within distance at most $(\frac{1}{4} - 2\tau) w(\partial B')$ from $V(\partial B')$ and the assumption that $C[p_i, p_{i+1}] \neq \partial B'[x_i, x_{i+1}]$.

Observe that $W_i = Q_i \cup \partial B'[x_i, x_{i+1}]$. We claim that:

$$w(W_i) \leq (1 - 2\tau)\, w(\partial B'). \tag{21}$$

If the paths P_i and P_{i+1} intersect, then $x_i = x_{i+1}$ by the construction of P_i and P_{i+1}, and (21) is immediate from (20) and the choice of τ. So assume that the paths P_i and P_{i+1} do not intersect. In particular, $x_i \neq x_{i+1}$. Let z_i be the vertex of $V(P_i) \cap V(C[p_i, p_{i+1}])$ that lies closest to x_i on P_i; define z_{i+1} similarly with respect to P_{i+1}. Observe that z_i lies closer to p_i on $C[p_i, p_{i+1}]$ than z_{i+1}, as otherwise $P_i[p_i, z_i]$ and $P_{i+1}[p_{i+1}, z_{i+1}]$ would intersect (recall that none of these paths contain an edge strictly enclosed by C). Hence, $C[z_i, z_{i+1}]$ is a subpath of $C[p_i, p_{i+1}]$. Let $Q'_i = P_i[x_i, z_i] \cup C[z_i, z_{i+1}] \cup P_{i+1}[z_{i+1}, x_{i+1}]$. Observe that Q'_i is a simple path of length at most $w(Q_i) \leq (\frac{1}{2} - 2\tau)w(\partial B')$ by (20). Moreover, the closed walk $W'_i := Q'_i \cup \partial B'[x_i, x_{i+1}]$ does not enclose any point strictly enclosed by C. Hence, $Q'_i \cup \partial B'[x_{i+1}, x_i]$ encloses the whole of C, and thus in particular the core face $f_{\texttt{core}}$. Thus, $(Q'_i, \partial B'[x_{i+1}, x_i])$ is not a (2τ)-carve, despite that $w(Q'_i) \leq \left(\frac{1}{2} - 2\tau\right) w(\partial B')$. Therefore, it must be that $w(\partial B'[x_{i+1}, x_i]) > \frac{1}{2}w(\partial B')$, and thus $w(\partial B'[x_i, x_{i+1}]) \leq \frac{1}{2}w(\partial B')$. Then (21) follows from (20).

It remains to observe that $\sum_{B_1 \in \mathcal{B}_i} w(\partial B_1) \leq w(W_i) \leq (1 - 2\tau)w(\partial B')$. □

Lemma 10.15. *If* `partition`(B') *recursively calls* `partition`(B^*), *then* $w(\partial B^*) \leq (1 - \tau/2) \cdot w(\partial B')$.

Proof. If a $(3 + 2\tau)$-short $\tau/2$-nice brick covering $\mathcal{B}$ has been found in B', then the claim follows from the niceness of $\mathcal{B}$. In the second case, when Theorem 7.1 is invoked, the claim follows by Lemma 10.14. □

Lemma 10.16. *There exists a universal constant C such that the following holds: for any call* `partition`(B'), *we have* $\sum_{B^* \in \mathcal{B}^r(B')} w(\partial B^*) \leq Cw(\partial B')$.

Proof. The claim is immediate for any $C \geq 3 + 2\tau$ in the case when a $(3 + 2\tau)$-short $\tau/2$-nice brick partition $\mathcal{B}$ has been found in B'. In the second case, when Theorem 7.1 is invoked, note that the claim follows for sufficiently large C by Lemma 10.14 and the bound of (19) that $s = \mathcal{O}(1)$. □

Lemma 10.17. *There exists a universal constant c such that the following holds: for any call* `partition`(B'), *in the subtree of the recursion tree rooted at this call there are at most*

$$c\left(\frac{\theta}{\varepsilon} \cdot \frac{w(\partial B')}{w(\partial B)}\right)^c$$

calls to `partition`(B^*) *where B^* is large (i.e., the call* `partition`(B^*) *does not finish after the first step).*

Proof. We prove the claim by induction, proceeding from the leaves to the root of the recursion tree. The claim is clearly true for any positive c if B' is tiny, as no recursive call is made.

Consider now a call `partition`(B') where B' is large. We use Lemmata 10.15 and 10.16; let C be the constant given by the latter. By the induction hypothesis, for sufficiently large c

that depends on $\tau = \frac{1}{36}$ and C, the number of calls in question is bounded by

$$\begin{aligned}
1 + \sum_{B^* \in \mathcal{B}^r(B')} & c\left(\frac{\theta}{\varepsilon} \cdot \frac{w(\partial B^*)}{w(\partial B)}\right)^c \\
&\leq 1 + c\frac{\theta^c}{\varepsilon^c} \sum_{B^* \in \mathcal{B}^r(B')} \frac{w(\partial B^*)}{w(\partial B)} (1-\tau)^{c-1} \left(\frac{w(\partial B')}{w(\partial B)}\right)^{c-1} \\
&\leq 1 + c\left(\frac{\theta}{\varepsilon} \cdot \frac{w(\partial B')}{w(\partial B)}\right)^c (1-\tau)^{c-1} \cdot C \\
&\leq c\left(\frac{\theta}{\varepsilon} \cdot \frac{w(\partial B')}{w(\partial B)}\right)^c .
\end{aligned}$$

The last inequality follows for sufficiently large c as

$$\left(\frac{\theta}{\varepsilon} \cdot \frac{w(\partial B')}{w(\partial B)}\right) > 1.$$

□

By applying Lemma 10.17 to the root call $\texttt{partition}(B)$ we obtain the following:

Corollary 10.18. *In the entire run of the algorithm there are at most* $\mathrm{poly}(\varepsilon^{-1}, \theta)$ *calls to* $\texttt{partition}(B')$ *where* B' *is large*

As a single call to $\texttt{partition}(B')$ takes $\mathcal{O}(|V(B')| \log |V(B')|)$ time, we have also that:

Corollary 10.19. *Phase one takes* $\mathrm{poly}(\varepsilon^{-1}, \theta)|B| \log |B|$ *time.*

We now bound the size and the length of the bricks in $\mathcal{A}$.

Lemma 10.20. *The sum of the lengths of the perimeters of all bricks in* $\mathcal{A}$ *is bounded by* $\mathrm{poly}(\varepsilon^{-1}, \theta) w(\partial B)$.

Proof. By Lemma 10.15, in each call $\texttt{partition}(B')$ we have $w(\partial B') \leq w(\partial B)$. Consider a call $\texttt{partition}(B')$ where B' is large. By Lemma 10.16, the sum of lengths of all perimeters of bricks $B^* \in \mathcal{B}^r(B')$ that are tiny (and hence will be inserted into $\mathcal{A}$) is bounded by $Cw(\partial B')$. Moreover, if Theorem 7.1 has been invoked, we have $w(\partial B^C) \leq \frac{16}{\tau^2} w(\partial B')$. Finally, by Corollary 10.18, there are at most $\mathrm{poly}(\varepsilon^{-1}, \theta)$ calls $\texttt{partition}(B')$ where B' is large. The lemma follows. □

Lemma 10.21. *The total number of edges and vertices in all bricks of* $\mathcal{A}$ *is bounded by* $\mathrm{poly}(\varepsilon^{-1}, \theta)|B|$.

Proof. Consider a call to $\texttt{partition}(B')$ where B' is large. First, observe that in this call at most one brick is put into $\mathcal{A}$. Moreover, observe that the total number of edges and vertices in all recursive calls $\texttt{partition}(B^*)$ for $B^* \in \mathcal{B}^r(B')$ is $\mathcal{O}(|B'|)$. Here we rely on the fact that in the algorithm of Theorem 9.2, each face of B' is contained in at most 7 bricks of $\mathcal{B}(B')$, and, if the algorithm of Theorem 7.1 has been invoked, then $\mathcal{B}(B')$ is a brick partition of B'. Finally, recall that if B' is tiny, then we simply put B' into $\mathcal{A}$. The bound of the lemma follows from Corollary 10.18. □

10.2.2 Phase two: constructing H from the decomposition

In the second phase we derive the output graph H from the brick covering $\mathcal{A}$.

Consider first a graph $H_0 := \bigcup_{B_1 \in \mathcal{A}} \partial B_1$. By Lemma 10.12, H_0 is connected and contains ∂B. Pick any finite face f of H_0. As H_0 is connected, the interior of f is homeomorphic to an open disc. Moreover, since H_0 is a union of simple cycles, there is no bridge in H_0 and, hence, each edge of H_0 appears on the boundary of f at most once (but H_0 may have articulation points, and one vertex may appear multiple times on the boundary of f).

Let C^f be the walk in B around the boundary of f and let G^f be the subgraph of B consisting of all edges of B that lie in f or on the boundary of f (i.e., all edges of B that are enclosed by C^f). Moreover, construct a brick B^f from G^f by 'straightening' the boundary C^f, that is, for each appearance of a vertex v on C^f, make a separate copy of v adjacent to all edges that were adjacent to this appearance. Observe that there is a natural homomorphism π^f from B^f to G^f that is bijective on the edge set of B^f and surjective on the vertex set.

For each brick B^f, apply Theorem 10.1 to obtain a graph H^f. Output $H := \bigcup_f \pi^f(H^f)$, where the union ranges over all finite faces of H_0. It remains to show that H has the properties desired by Theorem 10.11 and can be computed in the desired time.

As $\partial B^f \subseteq H^f$ for each face f, we have that $C^f \subseteq H$ for each f and, consequently, $\partial B \subseteq H$. By Theorem 10.1 and Lemma 10.20, there is a universal constant γ such that:

$$\begin{aligned}
w(H) &\leq \sum_f w(H^f) \\
&\leq \sum_f \gamma\varepsilon^{-5} w(\partial B^f) \\
&= \sum_f \gamma\varepsilon^{-5} w(C^f) \\
&\leq \gamma\varepsilon^{-5} 2w(H_0) \\
&\leq 2\gamma\varepsilon^{-5} \sum_{B_1 \in \mathcal{A}} w(\partial B_1) \\
&\leq 2\gamma\varepsilon^{-5} \cdot \mathrm{poly}(\varepsilon^{-1}, \theta) w(\partial B) \\
&\leq \mathrm{poly}(\varepsilon^{-1}, \theta) w(\partial B).
\end{aligned}$$

Therefore, $w(H)$ satisfies the desired bound.

The following lemma shows that H preserves approximate Steiner forests for any choice of terminal pairs on the perimeter of B.

Lemma 10.22. *For every set $\mathcal{S} \subseteq V(\partial B) \times V(\partial B)$ of size at most θ, there exists a Steiner forest F_H that connects $\mathcal{S}$ in H such that $w(F_H) \leq w(F_B) + 2\varepsilon w(\partial B)$ for any Steiner forest F_B that connects $\mathcal{S}$ in B.*

Proof. Let F_B be a Steiner forest connecting $\mathcal{S}$ in B of minimum possible length that additionally satisfies the properties promised by Lemma 10.13. We construct a subgraph $F_H \subseteq H$ connecting $\mathcal{S}$ of length at most $(1+\varepsilon)w(F_B) + \varepsilon w(\partial B)$. Since $w(F_B) \leq w(\partial B)$ (as ∂B connects $\mathcal{S}$), this would conclude the proof of the lemma.

First, construct a subgraph F as follows. Start with $F = F_B$. As long as there exists a vertex v that is of degree at least three in F and does not belong to $V(\partial B_1)$ for any $B_1 \in \mathcal{A}$, find any tiny $B_2 \in \mathcal{A}$ such that ∂B_2 strictly encloses v, delete from F all edges strictly enclosed by ∂B_2, add ∂B_2 instead, and take any spanning forest of the obtained graph. In this procedure we never introduce a vertex of degree at least three into F that does not belong to $V(H_0) =$

$\bigcup_{B_1 \in \mathcal{A}} V(\partial B_1)$, and hence such a tiny B_2 always exists by the properties of F_B promised by Lemma 10.13. Moreover, as $|\mathcal{S}| \leq \theta$, F_B contains at most θ vertices of degree at least three, and in the construction of F we made at most θ replacements. Consequently,

$$w(F) \leq w(F_B) + \theta \cdot \frac{\varepsilon}{\theta} w(\partial B) = w(F_B) + \varepsilon w(\partial B).$$

Consider the graph $F \setminus H_0$. Recall that F_B is a forest, $F \setminus F_B \subseteq H_0$ (in the process of constructing F we have only added edges of H_0 to F), and each vertex of degree at least three in F belongs to $V(H_0)$. Consider the following relation on the edge set of $F \setminus H_0$: two edges e_1, e_2 are in relation if and only if there exists a path in $F \setminus H_0$ that contains e_1 and e_2 and no internal vertex of this path belongs to $V(H_0)$. Observe that this is an equivalence relation. Moreover, as each vertex of degree at least three in F belongs to $V(H_0)$, each equivalence class in this relation is a path P that connects two vertices of $V(H_0)$, but all internal vertices of P do not belong to $V(H_0)$.

Let $\mathcal{P}$ be the family of equivalence classes of the aforementioned relation in $F \setminus H_0$. For each path $P \in \mathcal{P}$, proceed as follows. As no edge and no internal vertex of P belongs to H_0, there exists a finite face f of H_0 that contains P. Moreover, $(\pi^f)^{-1}(P)$ is a path in B^f, connecting two vertices of ∂B^f. By the properties of H^f (and in particular by Theorem 10.1), there exists a path Q in H^f connecting the same endpoints and of length at most $(1+\varepsilon)w((\pi^f)^{-1}(P)) = (1+\varepsilon)w(P)$. Hence, $\pi^f(Q)$ is a walk in G^f connecting the endpoints of P of length at most $(1+\varepsilon)w(P)$. To obtain a graph F_H, replace each P with $\pi^f(Q)$ in the graph F.

By construction, $F_H \subseteq H$ and F_H connects $\mathcal{S}$. Moreover, as each path $P \in \mathcal{P}$ has been replaced by a path of length at most $(1+\varepsilon)w(P)$, we have that $w(F_H) \leq (1+\varepsilon)w(F_B) + \varepsilon w(\partial B)$. This concludes the proof of the lemma. □

Observe that the lemma obtains an additive error $2\varepsilon w(\partial B)$ instead of $\varepsilon w(\partial B)$. The error of Theorem 10.11 can be obtained by appropriately rescaling ε at the beginning of the algorithm.

Finally, observe that Lemma 10.21 ensures that H_0 can be computed in $\mathrm{poly}(\varepsilon^{-1}, \theta)|B|$ time, and, consequently, the graph H can be computed in in $\mathrm{poly}(\varepsilon^{-1}, \theta)|B| \log |B|$ time. This completes the proof of Theorem 10.11.

10.3 Wrap up

We now pipeline the mortar graph construction of Borradaile et al [12] with Theorem 10.11 to conclude the proof of Theorem 1.7. In the language of brick coverings, the mortar graph construction of [12] can be summarized as follows.

Theorem 10.23 ([12], in particular Theorem 10.7)**.** *Given a brick B and an accuracy parameter $\varepsilon > 0$, one can in* $\mathrm{poly}(\varepsilon^{-1})|B| \log |B|$ *time compute a brick partition $\mathcal{B}$ of B of total perimeter $(1 + 18\varepsilon^{-1})w(\partial B)$ such that the perimeter $\partial B'$ of each brick $B' \in \mathcal{B}$ can be partitioned into four paths $\mathbf{N}_{B'} \cup \mathbf{W}_{B'} \cup \mathbf{S}_{B'} \cup \mathbf{E}_{B'}$ (the so-called north, west, south, and east boundaries, appearing in this counter-clockwise order), such that:*

1. *the total length of all parts $\mathbf{W}_{B'}$ and $\mathbf{E}_{B'}$ in all bricks of $\mathcal{B}$ is bounded by $\varepsilon w(\partial B)$; and*
2. *for any subgraph $F \subseteq B'$ of a brick $B' \in \mathcal{B}$, there exists a subgraph $F' \subseteq B'$ with the following properties:*
   (a) *$w(F') \leq (1 + c_1\varepsilon)w(F)$ for some universal constant c_1;*
   (b) *there are at most $\alpha(\varepsilon^{-1}) = o(\varepsilon^{-5.5})$ vertices of $V(\mathbf{N}_{B'}) \cup V(\mathbf{S}_{B'})$ that are incident to an edge of F' that does not belong to $\mathbf{N}_{B'} \cup \mathbf{S}_{B'}$;*

(c) if two vertices of $V(\mathbf{N}_{B'}) \cup V(\mathbf{S}_{B'})$ are connected by F, then they are also connected by F'.

The algorithm of Theorem 1.7 for a given brick B' and accuracy parameter $\varepsilon > 0$ can now be described as follows. First, we compute the brick partition $\mathcal{B}$ of Theorem 10.23 for the parameter ε and brick B. Second, for each $B' \in \mathcal{B}$, we invoke Theorem 10.11 for the brick B', accuracy parameter $\varepsilon' := \varepsilon/(1 + 18\varepsilon^{-1})$ and bound $\theta = (\alpha(\varepsilon^{-1}) + 4)^2$. Let $H(B')$ be the obtained subgraph for the brick B'. We output $H = \bigcup_{B' \in \mathcal{B}} H(B')$.

It remains to prove that H has the properties desired by Theorem 1.7 and can be computed in the desired time. Clearly, $\partial B \subseteq H$. By the bounds of Theorem 10.11 and the fact that $\alpha(\varepsilon^{-1}) = o(\varepsilon^{-5.5})$ we have that $w(H) \leq \mathrm{poly}(\varepsilon^{-1}) w(B)$. Moreover, as $\mathcal{B}$ is a brick partition, all calls to the algorithm of Theorem 10.11 run in total in $\mathrm{poly}(\varepsilon^{-1})|B|\log|B|$ time, and the time bound of Theorem 1.7 follows. It remains to argue that H preserves approximate Steiner forests for terminals on the perimeter of B.

To this end, consider any $\mathcal{S} \subseteq V(\partial B) \times V(\partial B)$ and let F be a Steiner forest connecting $\mathcal{S}$ in B of minimum possible length. First, define $F_1 := F \cup \bigcup_{B' \in \mathcal{B}} \mathbf{W}_{B'} \cup \mathbf{E}_{B'}$ and observe that $w(F_1) \leq w(F) + \varepsilon w(\partial B)$ by point 1 of Theorem 10.23. Then, for each $B' \in \mathcal{B}$ proceed as follows. Let $F_1(B')$ be the subgraph of F_1 consisting of all edges strictly enclosed by $\partial B'$. Let $F_2(B')$ be the subgraph promised by point 2 of Theorem 10.23 for the subgraph $F_1(B') \cup \mathbf{W}_{B'} \cup \mathbf{E}_{B'}$ of B'. Define

$$F_2 = \left(F_1 \setminus \bigcup_{B' \in \mathcal{B}} F_1(B') \right) \cup \bigcup_{B' \in \mathcal{B}} F_2(B').$$

By Theorem 10.23, we have

$$w(F_2(B')) \leq (1 + c_1\varepsilon)(w(F_1(B')) + w(\mathbf{W}_{B'}) + w(\mathbf{E}_{B'})).$$

Hence, for some universal constant c_2,

$$w(F_2) \leq (1 + c_1\varepsilon) w(F_1) + (1 + c_1\varepsilon)\varepsilon w(\partial B) \leq w(F_1) + c_2 \varepsilon w(\partial B).$$

Observe that $\mathbf{W}_{B'}, \mathbf{E}_{B'} \subseteq F_2$ for any $B' \in \mathcal{B}$. For each $B' \in \mathcal{B}$, we now proceed as follows. Define $F_2'(B')$ to be the subgraph of F_2 consisting of all edges strictly enclosed by $\partial B'$; observe that $F_2'(B') \subseteq F_2(B')$. Define $\mathcal{S}(B')$ to be the set of pairs (x, y) for which $x, y \in V(\mathbf{N}_B) \cup V(\mathbf{S}_B)$, $x \neq y$, and x, y are in the same connected component of $F_2'(B') \cup \mathbf{W}_{B'} \cup \mathbf{E}_{B'}$. Observe that if $(x, y) \in \mathcal{S}(B')$, then x (and similarly y) is an endpoint of $\mathbf{N}_{B'}$, an endpoint of $\mathbf{S}_{B'}$ or an endpoint of an edge of $F_2'(B') \subseteq F_2(B')$ that is strictly enclosed by $\partial B'$. By Theorem 10.23 and our choice of θ, $|\mathcal{S}(B')| \leq \theta$. Hence, by Theorem 10.11, there exists a subgraph $F_3(B')$ that connects $\mathcal{S}(B')$ in B', is contained in $H(B')$, and is of length

$$w(F_3(B')) \leq w(F_2'(B')) + w(\mathbf{W}_{B'}) + w(\mathbf{E}_{B'}) + \frac{\varepsilon}{1 + 18\varepsilon^{-1}} w(\partial B').$$

Define

$$F_3 = \left(F_2 \setminus \bigcup_{B' \in \mathcal{B}} F_2'(B') \right) \cup \bigcup_{B' \in \mathcal{B}} F_3(B').$$

As $\sum_{B' \in \mathcal{B}} w(\partial B') \leq (1 + 18\varepsilon^{-1}) w(\partial B)$ and $\sum_{B' \in \mathcal{B}} w(\mathbf{W}_{B'}) + w(\mathbf{E}_{B'}) \leq \varepsilon w(\partial B)$, we have that $w(F_3) \leq w(F) + c_3 \varepsilon w(\partial B)$ for some universal constant c_3. Moreover, by construction $F_3 \subseteq H$.

We now argue that F_3 connects $\mathcal{S}$. As F connects $\mathcal{S}$, so does F_1. To analyse F_2 and F_3, we introduce the following notion: for any $B' \in \mathcal{B}$ and $x \in V(\partial B')$, we set $\widehat{x}$ to be the common endpoint of $\mathbf{N}_{B'}$ and $\mathbf{W}_{B'}$ if $x \in V(\mathbf{W}_{B'})$, the common endpoint of $\mathbf{N}_{B'}$ and $\mathbf{E}_{B'}$ if $x \in V(\mathbf{E}_{B'})$, and $\widehat{x} = x$ otherwise. Observe that if $x, y \in V(\partial B')$ are connected by $F_1(B')$, then $\widehat{x}$ and $\widehat{y}$ are connected by $F_1(B') \cup \mathbf{W}_{B'} \cup \mathbf{E}_{B'}$ and, consequently, $\widehat{x}$ and $\widehat{y}$ are also connected by $F_2(B')$. Moreover, an identical claim is true for $F_1(B')$ replaced by $F_2'(B')$ and $F_2(B')$ replaced by $F_3(B')$. As all west and east boundaries of all bricks of $\mathcal{B}$ belong to F_1, F_2 and F_3, we infer that F_3 indeed connects $\mathcal{S}$. By taking ε/c_3 instead of ε at the beginning of the algorithm, Theorem 1.7 follows.

11 Applications: Planar Steiner Tree, Planar Steiner Forest and Planar Edge Multiway Cut

In this section we apply Theorem 1.1 to obtain polynomial kernels for Planar Steiner Tree, Planar Steiner Forest (parameterized by the number of edges in the tree or forest) and Planar Edge Multiway Cut (parameterized by the size of the cutset). The applications to Planar Steiner Tree and Planar Steiner Forest are rather straightforward, and rely on the trick from the EPTAS [12] to cut the graph open along an approximate solution. For Planar Edge Multiway Cut we need some more involved arguments to bound the diameter of the dual of the input graph, before we apply Theorem 1.1.

In all aforementioned problems, we consider the — maybe more practical or natural — optimization variants of the problem, instead of the decision ones. That is, we assume that the algorithm does not get the bound on the required tree, forest or cut, but instead is required to kernelize the instance with respect to the (unknown) optimum value. However, note that in all three considered problems an easy approximation algorithm is known, and the output of such an algorithm will be sufficient for our needs.

We also note that we do not care much about optimality of the exponents in the sizes of the kernels, as any application Theorem 1.1 immediately raises the exponents to the magnitude of hundreds. The main result of our work is the existence of polynomial kernels, not the actual sizes.

11.1 Planar Steiner Tree and Planar Steiner Forest

For both problems, we can apply the known trick of cutting open the graph along an approximate solution [12], which when combined with Theorem 1.1 gives the kernel.

Theorem 11.1 (Theorem 1.2 repeated). *Given a* Planar Steiner Tree *instance (G, S), one can in $\mathcal{O}(k_{OPT}^{142}|G|)$ time find a set $F \subseteq E(G)$ of $\mathcal{O}(k_{OPT}^{142})$ edges that contains an optimal Steiner tree connecting S in G, where k_{OPT} is the size of an optimal Steiner tree.*

Proof. We first manipulate the graph such that all terminals lie on the outer face. To do this, we find a 2-approximate Steiner tree T_{apx} for S in G in the following way. We run a breadth-first search in G from each terminal in S to determine a shortest path between each pair of the terminals. This takes $\mathcal{O}(|S||G|) = \mathcal{O}(k_{OPT}|G|)$ time. Define an auxiliary complete graph G' over S, where the length of an edge between two terminals is the length of the shortest path between these two terminals that we computed earlier. We then compute a minimum spanning tree in G'. This tree induces a Steiner tree in G, which is 2-approximate. Note that $k_{OPT} \leq |T_{apx}| \leq 2k_{OPT}$.

We now cut the plane open along tree T_{apx}, cf. [12] (see Figure 11). That is, we create an Euler tour of T_{apx} that traverses each edge twice in different directions, and respects the plane embedding of T_{apx}. Then we duplicate every edge of T_{apx}, replace each vertex v of T_{apx} with $d-1$ copies of v, where d is the degree of v in T_{apx}, and distribute the copies in the plane embedding so that we obtain a new face F whose boundary corresponding to the aforementioned Euler

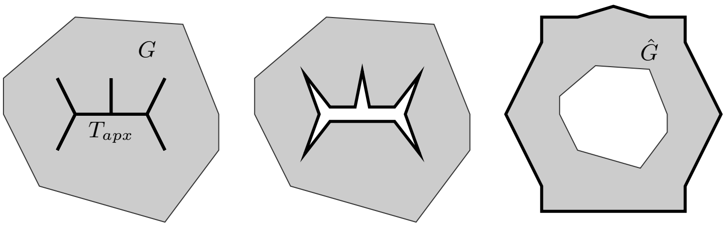


Figure 11: (Figure 2 repeated) The process of cutting open the graph G along the tree T_{apx}.

tour. Then fix an embedding of the resulting graph $\hat{G}$ that has F as its outer face. Observe that there exists a natural mapping π from $E(\hat{G})$ to $E(G)$, i.e., edges in $\hat{G}$ are mapped to edges from which they where obtained. Moreover, note that the terminals S lie only on the outer face of $\hat{G}$, and that $|\partial\hat{G}| \leq 4k_{OPT}$.

Finally, we obtain the kernel. Apply Theorem 1.1 to $\hat{G}$ to obtain a subgraph $\hat{H}$, which has size $\mathcal{O}(|\partial\hat{G}|^{142}) = \mathcal{O}(k_{OPT}^{142})$. Let $F = \pi(\hat{H})$. We show that F is a kernel for (G, S). Clearly, $|\pi(\hat{H})| \leq |\hat{H}| \leq \mathcal{O}(k_{OPT}^{142})$. Let T be an optimal Steiner tree in G for S and consider $\pi^{-1}(T)$. If $\pi^{-1}(T)$ contains edges e' and e'' for which there exists an edge $e \in G$ such that $\pi(e') = \pi(e'') = e$, then arbitrarily remove either e' or e''. Let $\hat{T}$ denote the resulting graph. By construction, $|T| = |\hat{T}|$. Observe that any connected component C of $\hat{T}$ is a connector for $V(C) \cap V(\partial\hat{G})$. Hence, there exists an optimal Steiner tree T_C in $\hat{H}$ that connects $V(C) \cap V(\partial\hat{G})$. Let $\hat{T}_H$ be the graph that is obtained from $\hat{T}$ by replacing C with T_C for each connected component of $\hat{T}$. Observe that during each such replacement, $\pi(\hat{T}_H)$ remains connected, because T was connected. Again, by construction, $|\hat{T}_H| \leq |\hat{T}|$. Now observe that $\pi(\hat{T}_H)$ is a subgraph of $\pi(\hat{H})$ connecting S in G, of not higher cost than T. □

For Planar Steiner Forest, we need to slightly preprocess the input instance, removing some obviously unnecessary parts, to bound the diameter of each connected component.

Theorem 11.2 (Theorem 1.3 repeated). *Given a* Planar Steiner Forest *instance* $(G, \mathcal{S})$, *one can in* $\mathcal{O}(k_{OPT}^{710}|G|)$ *time find a set* $F \subseteq E(G)$ *of* $\mathcal{O}(k_{OPT}^{710})$ *edges that contains an optimal Steiner forest connecting* $\mathcal{S}$ *in* G, *where* k_{OPT} *is the size of an optimal Steiner forest.*

Proof. Let $(G, \mathcal{S})$ be a Planar Steiner Forest instance. A forest with k_{OPT} edges has at most $2k_{OPT}$ vertices, and thus $|\mathcal{S}| = \mathcal{O}(k_{OPT}^2)$. We construct an approximate solution T_1, by taking a union of shortest s_1s_2-paths for all $(s_1, s_2) \in \mathcal{S}$. Clearly, $k_{OPT} \leq |T_1| \leq |\mathcal{S}|k_{OPT} = \mathcal{O}(k_{OPT}^3)$. Let $k_1 = |T_1|$.

We remove from G all vertices (and incident edges) that are at distance more than k_1 from all terminals of $\mathcal{S}$. Clearly, no such vertices or edges are used in a minimal solution for $(G, \mathcal{S})$ with at most k_1 edges.

Consider each connected component of G separately. Let G_0 be a component of G and let S_0 be the family of terminals of $\mathcal{S}$ in G_0. In $\mathcal{O}(|S_0| \cdot |G_0|)$ time, we construct a 2-approximate Steiner *tree* T_0 connecting S_0 in G_0. Note that, as each vertex of G_0 is within a distance at most k_1 from S_0, we have $|T_0| = \mathcal{O}(|S_0|k_1)$. As in the proof of Theorem 11.1, cut the graph G_0 open along T_0, obtaining a brick $\hat{G}_0$ of perimeter $|\partial\hat{G}_0| = \mathcal{O}(|S_0|k_1)$. Then apply the algorithm of Theorem 1.1 to $\hat{G}_0$, obtaining a subgraph $\hat{H}$. Finally, put the edges of G_0 that correspond to $\hat{H}$ into the constructed subgraph F. By similar arguments as in the proof of Theorem 11.1, F contains a minimum Steiner forest for $(G, \mathcal{S})$. The time bound and the bound on $|F|$ follows from the bound $k_1 = \mathcal{O}(k_{OPT}^3)$ and the fact that the union of all sets S_0 has size $2|\mathcal{S}| = \mathcal{O}(k_{OPT}^2)$. □

We observe that the size of the kernel can be improved to $\mathcal{O}(k_{OPT}^{426})$ by running a constant-factor approximation algorithm for Planar Steiner Forest to construct the forest T_1. However, when using the EPTAS for Planar Steiner Forest [32], this makes the algorithm run in $\mathcal{O}(k_{OPT}^{426}|G| + |G|\log^3|G|)$ time, which is no longer linear in $|G|$.

Another observation is that the size of the kernel can be improved if we consider a 'classic' kernel. That is, a kernel for the decision variant of the problem: does the planar graph G have a Steiner forest of size at most k? Then we can use k instead of k_1 in the above proof and return a kernel of size $\mathcal{O}(k^{426})$ in $\mathcal{O}(k^{426}|G|)$ time.

11.2 Planar Edge Multiway Cut

We are left with the case of Planar Edge Multiway Cut.

Theorem 11.3 (Theorem 1.4 repeated)**.** *Given a* Planar Edge Multiway Cut *instance (G, S), one can in polynomial time find a set $F \subseteq E(G)$ of $\mathcal{O}(k_{OPT}^{568})$ edges that contains an optimal solution to (G, S), where k_{OPT} is the size of this optimal solution.*

The idea of the kernel is that the PEMwC problem is some sort of Steiner Forest-like problem in G^*, the dual of G. However, to apply Theorem 1.1, we need to cut G^* open so that Theorem 1.1 can be applied to the brick created by this cutting. To bound the perimeter of this brick, it suffices to bound the diameter of G^*. This is done in Section 11.2.2, via a separate reduction rule. Earlier, in Section 11.2.1, we perform a few (well-known) regularization reductions on the input graph. Finally, in Section 11.2.3, we show formally how to cut open G^* and apply Theorem 1.1 to obtain the promised kernel.

Note that, contrary to the case of Planar Steiner Tree and Planar Steiner Forest, the preprocessing for PEMwC takes superlinear time, in terms of $|G|$.

In the rest of this section we assume that (G, S) is an input to PEMwC that we aim to kernelize. Note that, contrary to the previous sections, G may contain multiple edges. We fix some planar embedding of G, where multiple edges are drawn in parallel in the plane, without any other element of G between them.

In the course of the kernelization algorithm, we may perform two types of operations on G. First, if we deduce for some $e \in E(G)$ that there exists a minimum solution X not containing e, then we may contract e in G. During this contraction, any self-loops are removed, but multiple edges are kept. This operation is safe, because if F is a subgraph of G/e that has the properties promised by Theorem 11.3, then the projection of F into G satisfies those same properties. Second, if we deduce for some edge e that some minimum solution X to PEMwC on (G, S) contains e, we may delete e from G, analyze $G \setminus \{e\}$ obtaining a set F, and return $F \cup \{e\}$. As the size of the minimum solution to PEMwC decreases in $G \setminus e$, the size of F satisfies the bound promised in Theorem 1.1. Note that both edge contractions and edge deletions preserve planarity of G.

In the course of the arguments, we provide a number of reduction rules. At each step, the lowest-numbered applicable rule is used.

11.2.1 Preliminary reductions

In this section, we provide a few reduction rules to clean up the instance.

Reduction Rule 11.1. *If there is an edge e that connects two terminals, then delete e and include it into the constructed set F.*

Reduction Rule 11.2. *If $|S| \leq 1$, then return $F = \emptyset$.*

Now, we take care of the situation when the input instance (G, S) is in fact a union of a few PEMwC instances.

Reduction Rule 11.3. *If $G \setminus S$ is not a connected graph, then consider each of its connected component separately. That is, if $C_1, C_2, \ldots, C_s$ are connected components of $G \setminus S$, separately run the algorithm on instances $I_i = (G[C_i \cup N_G(C_i)], N_G(C_i))$ for $i = 1, 2, \ldots, s$, obtaining sets $F_1, F_2, \ldots, F_s$. Return $F = \bigcup_{i=1}^{s} F_i$.*

To see that Rule 11.3 is safe, first note that since $G[S]$ is edgeless (as Rule 11.1 has been performed exhaustively), the instances $(I_i)_{i=1}^{s}$ partition the edge set of G. Consequently, any path connecting two terminals in G, without any internal vertex being a terminal, is completely contained in one instance I_i. Hence, a minimum solution to (G, S) is the union of minimum solutions to the instances $(I_i)_{i=1}^{s}$, and thus if k_{OPT} is the size of an minimum solution to (G, S) and $k_{i,OPT}$ is the size of an minimum solution to I_i, then $k_{OPT} = \sum_{i=1}^{s} k_{i,OPT}$. Moreover, if $|F_i| \leq ck_{i,OPT}^{568}$ for some constant $c > 0$, then $|F| \leq ck_{OPT}^{568}$, as the function $x \mapsto cx^{568}$ is convex.

Therefore, in the rest of this section we may assume that $G \setminus S$ is connected.

We now introduce some notation with regards to cuts in a graph. For two disjoint subsets $A, B \subseteq V(G)$ we say that $X \subseteq E(G)$ is a *(A, B)-cut* if no connected component of $G \setminus X$ contains both a vertex of A and a vertex of B. For $A = \{a\}$ or $B = \{b\}$ we shorten this notion to (a, b)-cut, (a, B)-cut and (A, b)-cut. An (A, B)-cut X is *minimal* if no proper subset of X is an (A, B)-cut, and *minimum* if $|X|$ is minimum possible. For $X \subseteq E(G)$ and $A \subseteq V(G)$ we define $\texttt{reach}(A, X)$ as the set of those vertices $v \in V(G)$ that are contained in a connected component of $G \setminus X$ with at least one vertex of A. Note that X is a (A, B)-cut if and only if $\texttt{reach}(A, X) \cap \texttt{reach}(B, X) = \emptyset$, and X is a minimal (A, B)-cut if additionally each edge of X has one endpoint in $\texttt{reach}(A, X)$, and second endpoint in $\texttt{reach}(B, X)$. For a vertex t, we write $\texttt{reach}(t, X)$ instead of $\texttt{reach}(\{t\}, X)$. For any $Q \subseteq V(G)$, we define $\delta(Q)$ as the set of edges of G with exactly one endpoint in Q. Note that if $A \subseteq Q$ and $B \cap Q = \emptyset$, then $\delta(Q)$ is a (A, B)-cut. Moreover, if X is a (A, B)-cut then $\delta(\texttt{reach}(A, X)) \subseteq X$ and if X is a minimal (A, B)-cut then $\delta(\texttt{reach}(A, X)) = X$.

This section relies on the submodularity of the cut function $\delta()$:

Lemma 11.4 (submodularity of cuts [42]). *For any $P, Q \subseteq V(G)$ it holds that:*

$$|\delta(P)| + |\delta(Q)| \geq |\delta(P \cup Q)| + |\delta(P \cap Q)|.$$

From the submodularity of cuts we infer that if X and Y are minimum (A, B)-cuts, then $\delta(\texttt{reach}(A, X) \cup \texttt{reach}(A, Y))$ and $\delta(\texttt{reach}(A, X) \cap \texttt{reach}(A, Y))$ are minimum (A, B)-cuts as well. Therefore, there exists a unique minimum (A, B)-cut K with inclusion-wise maximal $\texttt{reach}(A, K)$. We call this cut *the minimum (A, B)-cut furthest from A*. Moreover, this cut can be computed in polynomial time (see for example [57]).

The submodularity of cuts also yields the following known reduction rule (cf. [17]).

Reduction Rule 11.4. *For all $t \in S$, let K_t be the minimum $(t, S \setminus \{t\})$-cut furthest from t. If $K_t \neq \delta(t)$ for some $t \in S$, then contract all edges with both endpoints in $\texttt{reach}(t, K_t)$ (i.e., contract $\texttt{reach}(t, K_t)$ onto t).*

Clearly, Reduction 11.4 can be applied in polynomial time. Note that if this rule is not applicable, then $\delta(\hat{t})$ is the unique minimum $(\hat{t}, S \setminus \{\hat{t}\})$-cut. For completeness, we provide the proof of its safeness.

Lemma 11.5. *Let K_t be the minimum $(t, S \setminus \{t\})$-cut furthest from t. Then there exists a minimum solution to (G, S) that does not contain any edge with both endpoints in $\texttt{reach}(t, K_t)$.*

Proof. Let X be a minimum solution of (G, S). Let $P = \texttt{reach}(t, X)$ and $Q = \texttt{reach}(t, K_t)$. Note that $P \cap S = \{t\}$ and, consequently, $\delta(P)$ is a $(t, S \setminus \{t\})$-cut. By submodularity of the cuts,

$|\delta(P\cup Q)|+|\delta(P\cap Q)| \leq |\delta(P)|+|\delta(Q)|$. As K_t is a minimum $(t, S\setminus\{t\})$-cut, $|\delta(P\cap Q)| \geq |\delta(Q)|$ and, consequently, $|\delta(P \cup Q)| \leq |\delta(P)|$. We infer that, if we define

$$Y := (X \setminus (E(G[Q]) \cup \delta(P))) \cup \delta(P \cup Q),$$

we have $|Y| \leq |X|$, as $\delta(P) \subseteq X$.

We claim that Y is a solution to (G, S); as $|Y| \leq |X|$ and $Q \subseteq \texttt{reach}(t, Y)$, this would finish the proof of the lemma. Assume otherwise, and let R be a path between two terminals in $G\setminus Y$. As X is a solution to (G, S), R contains an edge of $\delta(P)$ or a vertex of Q, and, consequently, contains a vertex of $P \cup Q$. Note that at least one endpoint of R is different than t; hence, R contains an edge of $\delta(P \cup Q)$, a contradiction, as $\delta(P \cup Q) \subseteq Y$. □

We now recall that the set of all minimum $t-(S\setminus\{t\})$ cuts is a 2-approximation for PEMwC (cf. [20]).

Lemma 11.6. *If Rule 11.4 is not applicable to (G, S), then $\bigcup_{t\in S} \delta(t)$ is a solution to (G, S) of size at most $2k_{OPT}$.*

Proof. Observe that $\bigcup_{t\in S} \delta(t)$ is indeed a solution. It remains to prove the bound. Let X be a solution to (G, S). Note that for each $t \in S$, the set $\delta(\texttt{reach}(t, X))$ is a $(t, S \setminus \{t\})$-cut in G. Consequently, $|\delta(\texttt{reach}(t, X))| \geq |\delta(t)|$. On the other hand, each edge $e \in X$ belongs to $\delta(\texttt{reach}(t, X))$ for at most two terminals $t \in S$. Hence,

$$2k_{OPT} = 2|X| \geq \sum_{t\in S} |\delta(\texttt{reach}(t, X))| \geq \sum_{t\in S} |\delta(t)| \geq \left|\bigcup_{t\in S} \delta(t)\right|,$$

and the lemma follows. □

We infer that, once Rule 11.4 is exhaustively applied, $k := |\bigcup_{t\in S} \delta(t)|$ satisfies $k_{OPT} \leq k \leq 2k_{OPT}$.

We now state the last clean-up rule.

Reduction Rule 11.5. *If there is an edge e of multiplicity larger than k, then contract e.*

11.2.2 Bounding the diameter of the dual

We are now ready to present a reduction rule that bounds the diameter of the dual of G. Recall that we assume that G is connected.

Arbitrarily, pick one terminal $\hat{t} \in S$. We construct a sequence of $(\hat{t}, S\setminus\{\hat{t}\})$-cuts $K_1, K_2, \ldots, K_r$ as follows. We start with $K_1 = \delta(\hat{t})$; recall that, once Rule 11.4 is not applicable, $\delta(\hat{t})$ is the unique minimum $(\hat{t}, S \setminus \{\hat{t}\})$-cut. Having constructed K_i, we proceed as follows. If there exists an edge in K_i that is not incident to a terminal in $S \setminus \{\hat{t}\}$, we pick one such edge uv arbitrarily and take K_{i+1} to be the minimum $(\texttt{reach}(\hat{t}, K_i) \cup \{u, v\}, S \setminus \{\hat{t}\})$-cut furthest from $\texttt{reach}(\hat{t}, K_i) \cup \{u, v\}$. Otherwise, we terminate the process. Note that the sequence $K_1, K_2, \ldots, K_r$ can be computed in polynomial time.

We note the following properties of the sequence $K_1, K_2, \ldots, K_r$.

Lemma 11.7. *If Rules 11.1–11.5 are not applicable, then the following holds:*

1. *$K_r = \bigcup_{t\in S\setminus\{\hat{t}\}} \delta(t)$;*

2. *$1 \leq |\delta(\hat{t})| = |K_1| < |K_2| < \ldots < |K_r| < 2k_{OPT}$;*

3. $r < 2k_{OPT}$*;*

4. for each $1 \leq i < r$*,* $\texttt{reach}(\hat{t}, K_i) \subsetneq \texttt{reach}(\hat{t}, K_{i+1})$*.*

Proof. We first show that when $K_i \neq \bigcup_{t \in S \setminus \{\hat{t}\}} \delta(t)$, for some i, then $K_{i+1} \neq K_i$. As $\bigcup_{t \in S \setminus \{\hat{t}\}} \delta(t)$ is a $(\hat{t}, S \setminus \{\hat{t}\})$-cut, and K_i is a minimal $(\hat{t}, S \setminus \{\hat{t}\})$-cut, we infer that there exists an edge $vt \notin K_i$ incident to a terminal $t \neq \hat{t}$. As Rule 11.3 is not applicable, $G \setminus S$ is connected and thus there exists a $\hat{t}v$-path Q, such that only the first edge of Q is incident to a terminal. We infer that Q intersects K_i, and K_i contains an edge not incident to $S \setminus \{\hat{t}\}$. Consequently, K_{i+1} can be constructed. This concludes the proof of the first claim.

For the second claim, note that K_i is the unique minimum $(\texttt{reach}(\hat{t}, K_i), S \setminus \{\hat{t}\})$-cut, thus $|K_{i+1}| > |K_i|$ for all $1 \leq i < r$. By Lemma 11.6, $|\bigcup_{t \in S} \delta(t)| \leq 2k_{OPT}$. As Rule 11.1 is not applicable, the sets $\delta(t)$ are pairwise disjoint. As Rules 11.2 and 11.3 are not applicable, $\delta(\hat{t}) \neq \emptyset$. We infer that

$$|K_r| = \left| \bigcup_{t \in S \setminus \{\hat{t}\}} \delta(t) \right| < \left| \bigcup_{t \in S} \delta(t) \right| \leq 2k_{OPT}.$$

The third claim follows directly from the second one, and the last claim is straightforward from the construction. □

The main claim of this section is the following.

Lemma 11.8. *Assume Rules 11.1–11.5 are not applicable to the* PEMwC *instance* (G, S)*. Moreover, assume there exists an edge* $e \in G$ *such that the distance, in the dual of* G*, between* e *and* $\bigcup_{i=1}^{r} K_i$ *is greater than* k*. Then there exists a minimum solution to* (G, S) *that does not contain* e*.*

Proof. Let X be a minimum solution to (G, S). If $e \notin X$, there is nothing to prove, so assume otherwise. As e is distant from $\bigcup_{i=1}^{r} K_i$, in particular $e \notin \bigcup_{i=1}^{r} K_i$. Recall that, since we assume G is connected,

$$\{\hat{t}\} = \texttt{reach}(\hat{t}, K_1) \subsetneq \texttt{reach}(\hat{t}, K_2) \subsetneq \ldots \subsetneq \texttt{reach}(\hat{t}, K_r) = V(G) \setminus (S \setminus \{\hat{t}\}).$$

Hence, there exists a unique index ι, $1 \leq \iota < r$, such that both endpoints of e belong to $\texttt{reach}(\hat{t}, K_{\iota+1}) \setminus \texttt{reach}(\hat{t}, K_\iota)$.

Consider now X as an edge subset of the dual of G, and let Y be the connected component of X that contains e. Let $S_Y = S \setminus \texttt{reach}(\hat{t}, Y)$, i.e., S_Y is the set of terminals separated in G from $\hat{t}$ by Y. Finally, we define $\overline{Y}$ to be the set of edges of G that are incident to a face of G that is incident to at least one edge of Y, i.e., the set of edges that are incident to the endpoints of Y in the dual of G.

We first claim the following.

Claim 11.9. $\overline{Y}$ *is a connected subgraph of* G*, disjoint from* $\bigcup_{i=1}^{r} K_i$ *and the endpoints of* $\overline{Y}$ *in* G *lie in* $\texttt{reach}(\hat{t}, K_{\iota+1}) \setminus \texttt{reach}(\hat{t}, K_\iota)$*.*

Proof. Since G is connected, the edges incident to a face of G form a closed walk, and, consequently, $\overline{Y}$ is a connected subgraph of G. As $Y \subseteq X$, $|Y| \leq |X| = k_{OPT} \leq k$. Hence, any face incident to an edge of Y is, in the dual of G, within distance less than k from a face incident to e. Consequently, by the definition of $\overline{Y}$ and the choice of e, $\overline{Y}$ cannot contain any edge of $\bigcup_{i=1}^{r} K_i$. By the connectivity of $\overline{Y}$, for any $1 \leq i \leq r$, $\overline{Y}$ is either fully contained in $G[\texttt{reach}(\hat{t}, K_i)]$ or fully contained in $G \setminus \texttt{reach}(\hat{t}, K_i)$. Hence, the last claim follows from the definition of ι. ⌟

Intuitively, Claim 11.9 asserts that Y is a connected part of the solution that lives entirely between K_ι and $K_{\iota+1}$. The role of Y in the solution X is to separate S_Y from $\hat{t}$ (and/or other terminals of $S \setminus S_Y$), and, possibly, separate some subsets of S_Y from each other. Define Z to be the set of those edges of $K_{\iota+1}$ whose endpoints are separated from $\hat{t}$ by Y, i.e., both do not belong to $\texttt{reach}(\hat{t}, Y)$. Note that, as $\overline{Y} \cap K_{\iota+1} = \emptyset$, for any $e' \in K_{\iota+1}$, either both endpoints of e' belong or both endpoints do not belong to $\texttt{reach}(\hat{t}, Y)$.

Claim 11.10. *$K := (K_{\iota+1} \setminus Z) \cup Y$ is a $(\hat{t}, S \setminus \{\hat{t}\})$-cut. Moreover, $\texttt{reach}(\hat{t}, K_\iota) \cup V(K_\iota \setminus K_{\iota+1}) \subseteq \texttt{reach}(\hat{t}, K)$.*

Proof. The second claim of the lemma is straightforward, as, by Claim 11.9, no edge of Y belongs to $K_\iota \setminus K_{\iota+1}$ nor does it have both endpoints in $\texttt{reach}(\hat{t}, K_\iota)$. For the first claim, assume the contrary, and let P be a $\hat{t}t$-path in $G \setminus K$ for some $t \in S \setminus \{\hat{t}\}$. As $K_{\iota+1}$ is a $(\hat{t}, t)$-cut, P contains an edge of Z. However, by the definition of Z, P contains an edge of Y, and P intersects K, a contradiction. ⌟

Recall now that $K_{\iota+1}$ is a minimum $(\texttt{reach}(\hat{t}, K_\iota) \cup \{u, v\}, S \setminus \hat{t})$-cut for some $uv \in K_\iota$. By Claim 11.10, K is also a $(\texttt{reach}(\hat{t}, K_\iota) \cup \{u, v\}, S \setminus \hat{t})$-cut. Hence, $|K| \geq |K_{\iota+1}|$ and, consequently, $|Y| \geq |Z|$.

We are now ready to make the crucial observation.

Claim 11.11. *The set $X' := (X \setminus Y) \cup Z$ is a solution to* PEMwC *on (G, S).*

Proof. Assume the contrary, and let P be a path connecting two terminals in $G \setminus X'$. We consider two cases, depending on whether there exists an endpoint of P that belongs to S_Y. If there exists such an endpoint, Z should substitute Y as a separator and should intersect P. Otherwise, Y does not play any substantial role in intersecting P as a part of the solution X, and $X \setminus Y$ should already intersect P. We now proceed with formal argumentation.

In the first case, assume that $t \in S_Y$ is an endpoint of P. As X is a solution to (G, S), P contains an edge of Y. Let uv be such an edge on P that is closest to t, where u lies before v on P. Note that $P[t, u]$ does not contain any edge of $K_{\iota+1}$: as $t \notin \texttt{reach}(\hat{t}, Y)$ and $P[t, u] \cap Y = \emptyset$, $P[t, u]$ is contained in $G \setminus \texttt{reach}(\hat{t}, Y)$ but $Z = K_{\iota+1} \cap (G \setminus \texttt{reach}(\hat{t}, Y))$ and P avoids Z. Recall that all endpoints of the edges of Y lie in $\texttt{reach}(\hat{t}, K_{\iota+1})$; hence, there exists a path Q connecting $\hat{t}$ with u that avoids $K_{\iota+1}$. Hence, $Q \cup P[u, t]$ is a $\hat{t}t$-path avoiding $K_{\iota+1}$, a contradiction to the definition of $K_{\iota+1}$.

In the second case, both endpoints of P belong to $\texttt{reach}(\hat{t}, Y)$. Denote them t_1 and t_2. As X is a solution to (G, S), P contains at least one edge of Y. Let e_1 be the first such edge, and let e_2 be the last one. Moreover, let v_1 be the endpoint of e_1 closer to t_1 on P and v_2 be the endpoint of e_2 closer to t_2 on P. Note that $v_1, v_2 \in \texttt{reach}(\hat{t}, Y)$, as both $P[t_1, v_1]$ and $P[t_2, v_2]$ do not contain any edge of Y. We also note that it may happen that $e_1 = e_2 = v_1v_2$, but $v_1 \neq v_2$ and v_1 is closer to t_1 on P than t_2. Observe that since $P[t_1, v_1]$ and $P[t_2, v_2]$ avoid both X' and Y, they also avoid X.

As Y is connected in the dual of G, there exists a unique face f_Y of $(G \setminus Y)[\texttt{reach}(\hat{t}, Y)]$, that contains Y. As $\texttt{reach}(\hat{t}, Y)$ is connected by definition and the interior of each face of a connected graph is isomorphic to an open disc (since we are working on the euclidean plane), the closed walk around f_Y in $\texttt{reach}(\hat{t}, Y)$ connects all vertices incident to Y that belong to $\texttt{reach}(\hat{t}, Y)$ and, by the definition of $\overline{Y}$, all edges of this closed walk belong to $\overline{Y} \setminus Y$. We infer that v_1 and v_2 lie in the same connected component of $\overline{Y} \setminus Y$.[4]

[4] Note that the argument of this paragraph fails if we assume only that G is embedded on, say, a torus, instead of a plane. We do not know how to fix it for graphs of higher genera.

By the definition of Y and $\overline{Y}$, we have $X \cap \overline{Y} = Y$. Hence, v_1 and v_2 lie in the same connected component of $G \setminus X$ and the same holds for t_1 and t_2 (via paths $P[t_1, v_1]$ and $P[t_2, v_2]$), a contradiction to the fact that X is a solution to (G, S). This finishes the proof of Claim 11.11. ⌟

Clearly, as $|Y| \geq |Z|$ and $Y \subseteq X$, we have $|X'| \leq |X|$. As $Z \subseteq K_{\iota+1}$, we have $e \notin X'$. Thus, by Claim 11.11, X' is a minimum solution to PEMwC on (G, S) that does not contain e. This concludes the proof of the lemma. □

Lemma 11.8 allows us to state the following reduction rule.

Reduction Rule 11.6. *Compute a choice of cuts $K_1, K_2, \ldots, K_r$ for some arbitrarily chosen $\hat{t} \in S$. If there exists an edge e in G whose distance from $\bigcup_{i=1}^{r} K_i$ in the dual of G is greater than k, contract e.*

Note that Rule 11.6 may be applied in polynomial time. Moreover, it bounds the diameter of the dual of G. To prove this claim, we need the following easy fact.

Lemma 11.12. *Let H be a connected graph, and let $D \subseteq V(H)$ be a subset of vertices such that every vertex of H is in distance at most r from some element of X. Then the diameter of H is bounded by $(2r+1)|D| - 1$.*

Proof. For a vertex $w \in V(H)$, let $\pi(w)$ be a vertex of D closest to v, breaking ties arbitrarily. For sake of contradiction assume that there exist two vertices $u, v \in V(H)$ such that the shortest path P in H between u and v is of length at least $(2r+1)|D|$. Then $|V(P)| \geq (2r+1)|D| + 1$, and by the pigeon-hole principle there must exist a vertex $x \in D$ such that $x = \pi(w)$ for at least $2r+2$ vertices of $V(P)$. Let w_1 be the first of these vertices and w_2 be the last; note that the distance between w_1 and w_2 on P is at least $2r+1$, since there are at least $2r$ vertices on P between them. Now obtain a walk P' by removing $P[w_1, w_2]$ from P, and inserting first a shortest path from w_1 to x and then a shortest path from x to w_2. By assumption, both these paths are of length at most r, so P' is shorter than P. This contradicts the minimality of P. □

We are ready to give a bound on the diameter of the dual of G.

Lemma 11.13. *If Rules 11.1–11.6 are not applicable, then the diameter of the dual of G is $\mathcal{O}(k_{OPT}^3)$.*

Proof. By Lemma 11.12, since the dual of G is connected, it suffices to identify a set D of $\mathcal{O}(k_{OPT}^2)$ vertices of G such that every vertex of G is in distance at most $k+1$ from D. We claim that $D = V(\bigcup_{i=1}^{r} K_i)$ is such a set. By Lemma 11.7 we have that $|D| \leq \mathcal{O}(k_{OPT}^2)$. Take now any vertex $v \in V(G)$ and, since Rules 11.2 and 11.3 are not applicable, let e be an arbitrary edge incident to v. Since Rule 11.6 is not applicable, e is in distance at most k from D, so also v is in distance at most $k+1$ from D. □

11.2.3 Cutting the dual open and applying Theorem 1.1

We now proceed to the application of Theorem 1.1. We start with the following observation.

Lemma 11.14. *If Rules 11.3 and 11.1 are not applicable, then each 2-connected component of $\bigcup_{t \in S} \delta(t)$ in the dual of G is a cycle. That is, $\bigcup_{t \in S} \delta(t)$ is a set of cacti in the dual of G.*

Proof. Let $H_0 = \bigcup_{t \in S} \delta(t)$ be a subgraph of the dual of G. First, note that if Rule 11.1 is not applicable, then $\delta(t)$, for $t \in S$, are edge-disjoint cycles in G^*. We claim that these cycles are precisely 2-connected components of H_0. For the sake of contradiction, assume that there exists a simple cycle C in H_0 that contains edges from cycles $\delta(t_1), \delta(t_2), \ldots, \delta(t_p)$, where $p \geq 2$. Since C is simple, we can assume that for each i, there exists an edge of $\delta(t_i)$ not contained in C. Let γ be the curve on the plane corresponding to cycle C. Observe that edges of G crossing γ are precisely the primal edges of C. Take t_1 and observe that in G there is an edge incident to t_1 crossing γ, and there is an edge incident to t_1 not crossing γ. Since Rule 11.1 is not applicable, we conclude that there exist nonterminal vertices on both sides of the curve γ. As each edge of H_0 is incident to a terminal, removing S from G disconnects nonterminal vertices on different sides of γ, and Rule 11.3 would be applicable. This is a contradiction. □

We now construct two subgraphs H_0 and H_s of the dual of G. Let $H_0 = \bigcup_{t \in S} \delta(t)$. We note that, by Lemma 11.14, for each connected component C of H_0, the closed walk around the outer face of C is an Eulerian tour of C – as shown on Figure 12 (a).

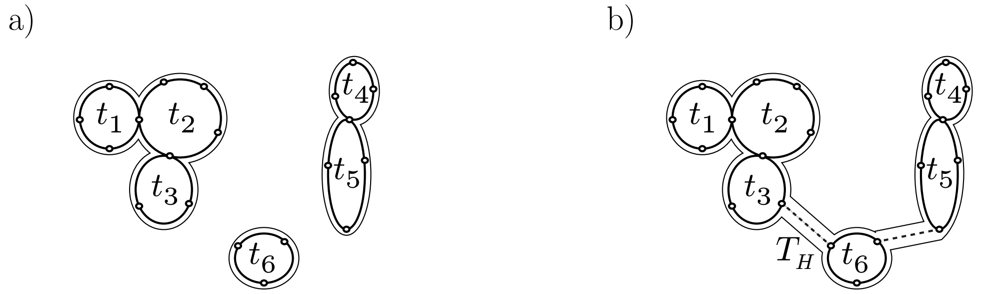


Figure 12: Panel a) shows the set of cacti together with their Eulerian tours, i.e., the graph H_0. Panel b) shows the construction of graph H_s, where the three Eulerian tours are jointed together using two copies of paths P and P'.

We now construct a connected subgraph H_s of the dual that contains as subgraph H_0. We first contract all connected components of H_0 to vertices, and find a minimum spanning tree T_H over these vertices (i.e., a 2-approximate Steiner tree). We set $H_s = H_0 \cup T_H$. Observe that $|H_0| = \sum_{t \in S} |\delta(t)| = k \leq 2k_{OPT}$. Moreover, by Lemma 11.13 the distance between any two terminals in the dual is bounded by $\mathcal{O}(k_{OPT}^3)$, so the cost of the MST T_H is bounded by $\mathcal{O}(k_{OPT}^4)$. We infer that $|H_s| = \mathcal{O}(k_{OPT}^4)$.

Now consider a multigraph H_{2s} obtained by taking a union of H_0 and two copies of T_H. We observe that H_{2s} is Eulerian, and let W be its Eulerian tour. Note that W is a closed walk around the outer face of H_s and each edge of H_0 appears exactly once on W and each edge of $H_s \setminus H_0 = T_H$ appears exactly twice on W. Hence, $|W| = \mathcal{O}(k_{OPT}^4)$. We cut the dual of G open along W. That is, we start with G^*, the dual of G, we duplicate each edge of $H_s \setminus H_0$ and, for each vertex $v \in V(H_s)$, we create a of copies of v equal to the number of appearances of v on W. Let $\hat{G}^*$ be the graph obtained in this way. In $\hat{G}^*$ the walk W becomes a simple cycle, enclosing a face f_W. We fix an embedding of $\hat{G}^*$ where f_W is the outer face. In this way $\hat{G}^*$ is a brick with perimeter of length $\mathcal{O}(k_{OPT}^4)$. Let π be a mapping that assigns to each edge of $\hat{G}^*$ its corresponding edge of G and G^*.

We apply Theorem 1.1 to the brick $\hat{G}^*$, obtaining a set F' of size $\mathcal{O}(k_{OPT}^{568})$. The set F' naturally projects to a set $F \subseteq E(G)$ via the mapping π. We claim that we may return the set F in our algorithm. That is, to finish the proof of Theorem 11.3 we prove the following lemma.

Lemma 11.15. *There exists a minimum solution X to* PEMwC *on (G, S) that is contained in F.*

Proof. Let X be a solution to PEMwC on (G, S) that minimizes $|X \setminus F|$. By contradiction, assume $X \setminus F \neq \emptyset$.

We define the following binary relation $\mathcal{R}$ on X: $\mathcal{R}(e, e')$ if and only if there exists a walk in G^* containing e and e', with all edges in X and all internal vertices not in $V(H_s)$. Clearly, $\mathcal{R}$ is symmetric and reflexive. We show that it is also transitive. Assume $\mathcal{R}(e, e')$ and $\mathcal{R}(e', e'')$, with witnessing paths P and P'. If $e = e'$ or $e' = e''$, the claim is obvious, so assume otherwise. We may assume that P starts with e and ends with e' and P' starts with e' and ends with e''. If P and P' traverse e' in the same direction then $P \cup P'$ is a witness to $\mathcal{R}(e, e'')$, as P and P' are of length at least two. In the other case, $(P \setminus e') \cup (P' \setminus e')$ is a witness to $\mathcal{R}(e, e'')$. Thus, $\mathcal{R}$ is an equivalence relation.

Note that any edge of $X \cap H_s$ is in a singleton equivalence class of $\mathcal{R}$. Let Y be the equivalence class of $\mathcal{R}$ that contains an element of $X \setminus F$. As $H_s \subseteq F$, we infer that $Y \cap H_s = \emptyset$ and, consequently, Y is also a subgraph (subset of edges) of $\hat{G}^*$. Let $\widehat{S} = V(\partial \hat{G}^*) \cap V(Y)$ in $\hat{G}^*$. We note that Y is a connected subgraph of $\hat{G}^*$ that connects $\widehat{S}$ – see Figure 13.

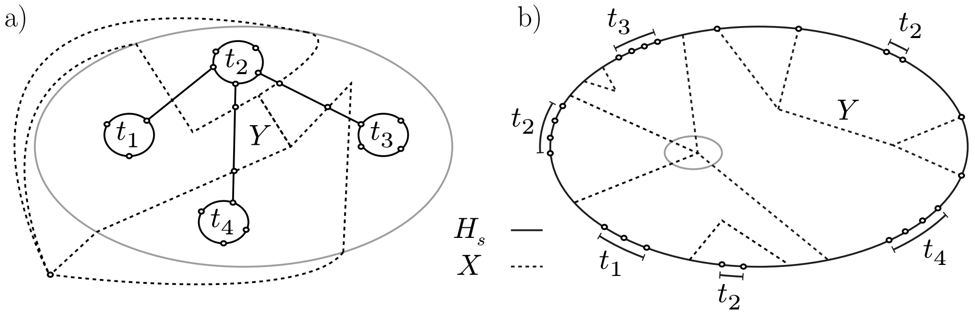


Figure 13: The figure shows the solution X to PEMwC and set Y in a) the dual graph G^* and b) the cut open dual graph $\hat{G}^*$.

By the properties of F', there exists a set $Z' \subseteq F'$ that connects $\widehat{S}$ in $\hat{G}^*$ and $|Z'| \leq |Y|$. Let $Z = \pi(Z') \subseteq F$. We claim that $X' := (X \setminus Y) \cup Z$ is a solution to (G, S) as well. This would contradict the choice of X, as $|X'| \leq |X|$ and $|X' \setminus F| < |X \setminus F|$.

So assume the contrary, and let P be a path connecting two terminals t^1 and t^2 in $G \setminus X'$. We may assume that P does not contain any terminal as an internal vertex. Note that P starts and ends with an edge of $H_0 \subseteq H_s$. As Rule 11.1 is not applicable, P is of length at least two. Let $e_0, e_1, e_2, \ldots, e_d$ be the edges of $P \cap H_s$, in the order of their appearance on P, let $e_i = u_i v_i$, where u_i lies closer on P to t^1 than v_i does. Note that $e_0, e_d \in H_0$ but $e_i \in H_s \setminus H_0$ for $1 \leq i < d$. Let $\hat{G}^{**}$ be the dual of $\hat{G}^*$. For each $i = 1, 2, \ldots, d$, we define a cycle Q_i in $\hat{G}^{**}$ as follows. Consider first path $P[u_{i-1}, v_i]$, and observe that every edge of this path apart from the first and the last is present in $\hat{G}^{**}$. Therefore, in $P[u_{i-1}, v_i]$ replace the edge e_{i-1} (belonging to H_s) with the copy of e_{i-1} in $\hat{G}^{**}$ that leads from the outer face of $\hat{G}^*$ to the face v_{i-1}, and replace the edge e_i with a copy of e_i in $\hat{G}^{**}$ that leads from u_i to the outer face of $\hat{G}^*$. Although Q_i is a cycle in $\hat{G}^{**}$, we call the aforementioned copy of e_{i-1} *the first arc* of Q_i, and the copy of e_i *the last arc*.

The set $\widehat{S}$ splits $\partial \hat{G}^*$ into a number of arcs $A_1, A_2, \ldots, A_{\max(1, |\widehat{S}|)}$. If, for some $1 \leq i \leq d$, the first and the last edge of the cycle Q_i lies in different arcs A_α and A_β, then Q_i intersects Z', and, consequently, P intersects Z, a contradiction to the choice of P – see Figure 14 (a). Hence, for all $1 \leq i \leq d$, the first and the last arc of Q_i lies in the same arc $A_{\alpha(i)}$. We now reach a contradiction by showing that t^1 and t^2 lie in the same connected component of $G \setminus X$.

As P avoids X' and $Y \cap H_s = \emptyset$, P avoids $X \cap H_s$ and $e_0, e_1, \ldots, e_d \notin X$. Let i be the smallest integer such that e_i does not lie in the same connected component of $G \setminus X$ as t^1. If

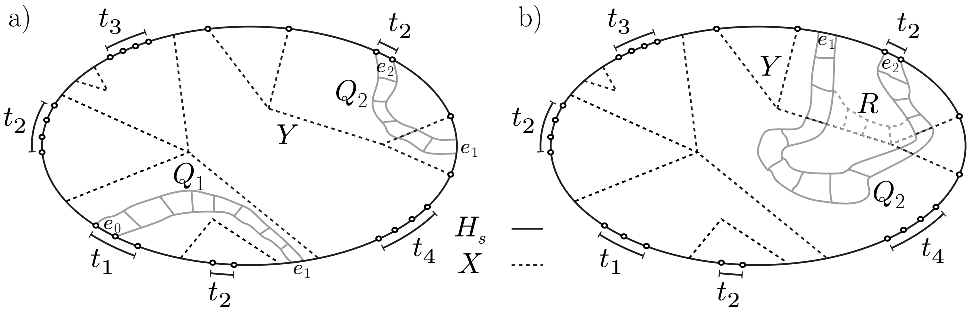


Figure 14: The path P in G can be seen as a sequence of faces in the dual. On panel (a) the last and the first edge of Q_2 lie in different arcs, whereas on (b) these edge belong to the same arc.

such i does not exist, the claim is proven as t^2 is an endpoint of e_d. Consider $P[v_{i-1}, u_i]$; note that this is also a subpath of Q_i, as it does not contain any edge of H_s. Recall that P avoids $X \setminus Y$. Hence, $P[v_{i-1}, u_i]$ intersects Y. Moreover, the first and the last edge of $P[v_{i-1}, u_i]$ lies on the same arc $A_{\alpha(i)}$, so $P[v_{i-1}, u_i]$ intersects Y at least twice. We treat now $P[v_{i-1}, u_i]$ as a subpath of Q_i, i.e., a path in $\hat{G}^{**}$. Let f_1 be the first face of $\hat{G}^*$ on $P[v_{i-1}, u_i]$ that is incident to an edge of Y, and let f_2 be the last such face. Observe that the prefix of $P[v_{i-1}, u_i]$ up to f_1 and the suffix of $P[v_{i-1}, u_i]$ from f_2 avoid both X' and Y, so they also avoid X.

Let us now show that there exists a path R in $\hat{G}^{**}$ connecting f_1 and f_2 that uses only edges that in $\hat{G}^*$ are incident to the endpoints of Y, but do not belong to Y nor $\partial\hat{G}^*$. Existence of path R can be inferred as follows. Take the set of faces $F_{\alpha(i)}$ of $\hat{G}^*$ that are reachable in $\hat{G}^{**}$ from edges of the arc $A_{\alpha(i)}$ without passing through the infinite face of $\hat{G}^*$, or traversing edges of Y. Consider also $F_{\alpha(i)}$ as a subset of plane obtained by gluing these faces together along all the edges between them that are not contained in Y. By the definition of Y as an equivalence class of $\mathcal{R}$, the boundary of $F_{\alpha(i)}$ is a closed walk that consists of arc $A_{\alpha(i)}$ and edges of Y that are incident to faces of $F_{\alpha(i)}$. By the definition, both of f_1 and f_2 are incident to the part of boundary of $F_{\alpha(i)}$ that is contained in Y. Path R can be then obtained by traversing faces of $F_{\alpha(i)}$ along its boundary, choosing the direction of the traversal so that part of the boundary of $F_{\alpha(i)}$ that is the arc $A_{\alpha(i)}$ is not traversed – see Figure 14 (b).

Since Y is an equivalence class of $\mathcal{R}$, edges of R do not belong to X (as otherwise they would be in relation with the edges of Y). Let R' be $P[v_{i-1}, u_i]$ with subpath between faces f_1 and f_2 replaced with R. If we now project R' to G^* and G using π, we infer that v_{i-1} and u_i lie in the same connected component of $G \setminus X$, a contradiction to the choice of i. This finishes the proof of the lemma, and concludes the proof of Theorem 11.3. □

12 Extending to bounded-genus graphs

In this section we extend the results from planar graphs to bounded genus graphs, using the framework of Borradaile et al. [11]. The idea is to reduce the bounded genus case to the planar case by cutting the graph embedded on a surface of bounded genus into a planar graph, using only a cutset of small size.

As in [11], we assume that we are given a combinatorial embedding of genus g of an input graph G, where the interior of each face is homeomorphic to an open disc. We proceed as in Sections 4.1 and 4.2 of [11]: given a brick embedded on a surface of genus g (i.e., a graph with a designated face), we may cut along a number of "short" cutpaths to make the brick planar. More precisely, the following theorem summarizes the results of [11] in our terminology, in particular the proved guarantees about the behaviour of procedures `Preprocess` and `Planarize` in [11].

Theorem 12.1 ([11], with adjusted terminology and parameter μ set to 1). *Let G be a connected graph embedded into a surface of genus g, and let $S \subseteq V(G)$ be a set of terminals in G. Let OPT be the weight of an optimum Steiner tree connecting S in G. Then one can in $\mathcal{O}(|G|)$ time find subgraphs CG and G' of G such that the following holds:*

- *$CG \subseteq G' \subseteq G$, CG and G' are connected, and CG contains all the terminals of S;*
- *all the vertices and edges of G' are at distance at most $4OPT$ from S in G', and G' contains all the vertices and edges of G that are at distance at most $2OPT$ from S in G;*
- *cutting G' along CG results in a planar graph G_p with the infinite face (corresponding to cut-open CG) being a simple cycle of length at most $8(2g + 2)OPT$.*

Let us remark that a combinatorial embedding of G' can be easily derived from a combinatorial embedding of G by removing all the vertices and edges not present in G', and replacing each new face whose interior ceased to be homeomorphic to an open disc with a number of disc faces.

By combining Theorem 12.1 with Theorem 1.1 we obtain the following.

Theorem 12.2 (Main Theorem for graphs of bounded genus). *Let B be a connected graph, with a combinatorial embedding into a surface of genus g. Let f be a simple face of B. Then one can find in $\mathcal{O}(|\partial f|^{142} \cdot (g + 1)^{142} \cdot |B|)$ time a subgraph $H \subseteq B$ such that*

(i) $\partial f \subseteq H$,

(ii) $|E(H)| = \mathcal{O}(|\partial f|^{142} \cdot (g + 1)^{142})$, and

(iii) for every set $S \subseteq V(\partial f)$, H contains some optimal Steiner tree in B connecting S.

Proof. Let $S_0 = V(\partial f)$. Observe that if OPT is the optimum weight of a Steiner tree connecting S_0 in B, then $OPT \leq |\partial f|$. We apply the algorithm of Theorem 12.1 to B, obtaining graphs B' and CB with the promised guarantees. Note that if B_p is the planar brick obtained from B' by cutting open along CB, then $|\partial B_p| \leq 8(2g + 2)|\partial f|$. The theorem now follows from an application of Theorem 1.1 to the brick B_p, and projecting the obtained subgraph $H_p \subseteq B_p$ back to B'. Note here that no edge of B that is not present in B' can participate in any optimum Steiner tree connecting any subset of S_0. □

Using Theorem 12.2 instead of Theorem 1.1, we immediately obtain bounded-genus variants of Theorems 11.1 and 11.2.

Theorem 12.3. *Given a* Steiner Tree *instance (G, S) together with an embedding of G into a surface of genus g where the interior of each face is homeomorphic to an open disc, one can in $\mathcal{O}(k_{OPT}^{142}(g+1)^{142}|G|)$ time find a set $F \subseteq E(G)$ of $\mathcal{O}(k_{OPT}^{142}(g+1)^{142})$ edges that contains an optimal Steiner tree connecting S in G, where k_{OPT} is the size of an optimal Steiner tree.*

Theorem 12.4. *Given a* Steiner Forest *instance $(G, \mathcal{S})$ together with an embedding of G into a surface of genus g where the interior of each face is homeomorphic to an open disc, one can in $\mathcal{O}(k_{OPT}^{710}(g+1)^{710}|G|)$ time find a set $F \subseteq E(G)$ of $\mathcal{O}(k_{OPT}^{710}(g+1)^{710})$ edges that contains an optimal Steiner forest connecting $\mathcal{S}$ in G, where k_{OPT} is the size of an optimal Steiner forest.*

We note that the arguments of Section 11.2 for Planar Edge Multiway Cut heavily rely on the planarity of the input graph, and the question of a polynomial kernel for Multiway Cut on graphs of bounded genus remains open.

We can plug the kernel given by Theorem 12.3 directly into the algorithm of Tazari [67] for Steiner Tree on graphs of bounded genus to obtain the following result:

Corollary 12.5. *Given a graph G with an embedding into a surface of genus g where the interior of each face is homeomorphic to an open disc, a terminal set $S \subseteq V(G)$, and an integer k, one can in $2^{\mathcal{O}_g(\sqrt{k \log k})} + \mathcal{O}(k_{OPT}^{142}(g+1)^{142}|G|)$ time decide whether the* Planar Steiner Tree *instance (G, S) has a solution with at most k edges.*

In this corollary, the hidden constant in $\mathcal{O}_g(\cdot)$ is some computable function of g.

13 Planar Edge Multiway Cut: Subexponential-Time Algorithm

In this section we show that the approach of Tazari for Steiner Tree [67] can be extended to Edge Multiway Cut.

Theorem 13.1. *Given a planar graph G, a terminal set $S \subseteq V(G)$, and an integer k, one can in $|G|^{\mathcal{O}(\sqrt{k})}$ time decide whether the* Planar Edge Multiway Cut *instance (G, S) has a solution with at most k edges.*

Proof. First, assume that (G, S, k) is a YES-instance and let X be an arbitrary minimum solution. We follow Baker's approach in G^*, the dual of G. Let f be an arbitrary vertex of G^*. Perform breadth-first search in G^*, starting from f, and let E_j, $j = 0, 1, 2, \ldots$ be the set of edges of G^* that connect the vertices of distance j from f with vertices of distance $(j+1)$. Note that the sets E_j are pairwise disjoint, but $\bigcup_j E_j$ may be a proper subset of $E(G^*)$. Denote $\ell = \lceil \sqrt{k} \rceil$. For $0 \leq i < \ell$, let $L_i = \bigcup_{j \geq 0} E_{i+j\ell}$. Branch into ℓ subcases, guessing an index $0 \leq i < \ell$ where $|X \cap L_i| \leq \sqrt{k}$. Furthermore, branch into $(\lfloor \sqrt{k} \rfloor + 1)$ subcases guessing $|X \cap L_i|$ and branch into at most $|V(G)|^{\lfloor \sqrt{k} \rfloor}$ subcases guessing the set $X \cap L_i$ itself. Label each branch with a pair (i, Y): the index of the layer L_i and the set $Y \subseteq L_i$ guessed (that is supposed to be $X \cap L_i$). Contract the edges of $L_i \setminus Y$ in the graph G (keeping multiple edges). Let H be the obtained graph.

We claim that after this operation the treewidth of H is bounded by $\mathcal{O}(\sqrt{k})$. By [14], it suffices to bound the treewidth of H^*, the dual of H. Recall that a contraction of an edge in a planar graph corresponds to a deletion of this edge in the dual. Hence, H^* is isomorphic to $G^* \setminus (L_i \setminus Y)$. However, each connected component of $G^* \setminus L_i$ is ℓ-outerplanar, and $|Y| \leq \sqrt{k}$. This finishes the proof of the treewidth bound of H^* and, consequently, of H.

To finish the proof of the theorem it suffices to note that a given Multiway Cut instance (G, S), equipped with a tree decomposition of G of width t, one can decide whether this instance has a solution of size at most k in $(|S|t)^{\mathcal{O}(t)}\text{poly}(|G|)$ time by a straightforward dynamic-programing routine[5]. Indeed, suppose we consider a bag B in the tree decomposition and we define $A \subseteq V(G)$ to be union of bags in the subtree rooted at B (including B itself). Then in a state of the dynamic-programing algorithm we need to remember the following information (F is a solution that conforms to the state): for each vertex $z \in B$, which terminal lies in the same connected component of $G[A] \setminus F$ as the vertex z, and how the vertices of B are

[5] We observe that this straightforward algorithm can be easily improved to a $t^{\mathcal{O}(t)}\text{poly}(|G|)$-time algorithm, since for a connected component intersecting the bag we do not need to remember precisely which terminal is contained in it, but only whether such a terminal exists or not. This running time can be further refined to $2^{\mathcal{O}(t)}\text{poly}(|G|)$ using the framework of sphere-cut decompositions and Catalan structures [28].

partitioned by the connected components of $G[A] \setminus F$. Since $|S| \leq |G|$ and $t \leq |G|$, this implies a $|G|^{\mathcal{O}(t)}$ algorithm. In our case t is $\mathcal{O}(\sqrt{k})$, which implies the theorem. □

By pipelining the kernelization algorithm of Theorem 1.4 with Theorem 13.1 we obtain the second claim of Corollary 1.5.

14 Planar Steiner Forest: No Subexponential-Time Algorithm

In this section, we prove Theorem 1.6, which states that no algorithm can decide in $2^{o(k)}\text{poly}(|G|)$ time whether Planar Steiner Forest instances $(G, \mathcal{S})$ have a solution with at most k edges, unless the Exponential Time Hypothesis fails. The Exponential Time Hypothesis was proposed by Impagliazzo, Paturi, and Zane [46]. Using the formulation by Fomin and Kratsch [38], it hypothesizes that no algorithm can decide instances of 3-SAT in $2^{o(n)}$ time, where n is the number of variables in the formula of the instance. Using the Sparsification Lemma [46], this is equivalent (see [38]) to the hypothesis that no algorithm can decide instances of 3-SAT in $2^{o(m)}$ time, where m is the number of clauses in the formula of the instance. It is this formulation of the Exponential Time Hypothesis that we rely on here.

To prove Theorem 1.6, we need a reduction from 3-SAT to Planar Steiner Forest. We use the following intermediate problem, which was also considered by Bateni et al. [5] in their NP-hardness reduction of Planar Steiner Forest on planar graphs of treewidth 3. Let the boolean relation $R(f, g, h)$ be equal to $(f = h) \vee (g = h)$. Then an *R-formula* is a conjunction of relations $R(f, g, h)$, where each of f, g, h can be a boolean variable, true (1), or false (0). For example, $R(x_1, x_2, x_3) \wedge R(x_1, 0, x_2) \wedge R(0, 1, x_3)$ is a valid R-formula. We explicitly mention here that it is critical that in $R(f, g, h)$ none of f, g, h can be the negation of a boolean variable. Then one can define the following problem:

R-SAT **Input:** An R-formula ϕ. **Task:** Decide whether ϕ is satisfiable.

Bateni et al. [5] essentially show the following result as part of their Theorem 8.2:

Lemma 14.1 ([5]). *Let ϕ be an R-formula on n variables and m clauses. Then in polynomial time one can construct an instance $(G_\phi, \mathcal{S}_\phi)$ of* Planar Steiner Forest *such that G_ϕ is a planar graph of treewidth 3, and $(G_\phi, \mathcal{S}_\phi)$ has a solution with at most $n + 3m$ edges if and only if ϕ is satisfiable.*

We can use this lemma to prove the following result, which is stronger than Theorem 1.6, and thus implies it.

Theorem 14.2. *If there is an algorithm that can decide in time $2^{o(k)}\text{poly}(|G|)$ whether* Planar Steiner Forest *instances $(G, \mathcal{S})$, where G has treewidth 3, have a solution with at most k edges, then the Exponential Time Hypothesis fails.*

Proof. Consider an instance of 3-SAT and let ψ be the CNF-formula of this instance. Let n denote the number of variables that appear in ψ and let m denote the number of clauses of ψ. Since each clause contains at most three variables, $m \geq n/3$ and thus $n \leq 3m$.

We first construct an R-formula ϕ that is equivalent to ψ. For each variable x_i ($i \in \{1, \dots, n\}$) that appears in ψ, add the *variable relations* $R(x_i^+, x_i^-, 1)$ and $R(x_i^+, x_i^-, 0)$ to ϕ. Here x_i^+ and x_i^- are new variables, which indicate whether x_i will be true or false respectively. Note that the relations ensure that $T'(x_i^+) \neq T'(x_i^-)$ for any truth assignment T' that satisfies both relations. Now consider a clause $C_j = (a \vee b \vee c)$ of ψ ($j \in \{1, \ldots, m\}$) — if C_j actually contains at most two literals, then we pretend that $c = 0$; if C_j contains one literal, then we also pretend that $b = 0$. Define a' as follows. If a is a variable x_i, then let $a' = x_i^+$. If a is the negation of a variable x_i, then let $a' = x_i^-$. Otherwise, i.e. if $a = 0$ or $a = 1$, then let $a' = a$. Define b' and c' similarly. Then, add to ϕ two new variables y_j^+ and y_j^-, and the following *clause relations*: $R(a', b', y_j^+)$, $R(0, c', y_j^-)$, and $R(y_j^+, y_j^-, 1)$. We claim that ψ is satisfiable if and only if ϕ is satisfiable.

Suppose that ψ is satisfiable, and let T be a satisfying truth assignment for ψ. We extend T to also cover negations of variables, i.e. $T(\neg x_i) = \neg T(x_i)$. We construct a satisfying truth assignment T' for ϕ as follows. If $T(x_i) = 1$, then let $T'(x_i^+) = 1$ and $T'(x_i^-) = 0$; otherwise, let $T'(x_i^+) = 0$ and $T'(x_i^-) = 1$. This satisfies all variable relations. Consider any clause $C_j = (a \vee b \vee c)$ of ψ. If $T(a) = 1$ or if $T(b) = 1$, then set $T'(y_j^+) = 1$ and $T'(y_j^-) = 0$. Otherwise, i.e. if $T(a) = 0$ and $T(b) = 0$, then $T(c) = 1$, and set $T'(y_j^+) = 0$ and $T'(y_j^-) = 1$. This satisfies all clause relations of ϕ. Hence, T' is a satisfying truth assignment for ϕ.

Suppose that ϕ is satisfiable, and let T' be a satisfying truth assignment for ϕ. We construct a satisfying truth assignment T for ψ as follows: set $T(x_i) = T'(x_i^+)$ for each variable in ψ. Again, we extend T to also cover negations of variables, i.e. $T(\neg x_i) = \neg T(x_i)$. Consider any clause $C_j = (a \vee b \vee c)$ of ψ. If $T'(y_j^-) = 1$, then it follows from the clause relations that $T(c) = 1$. Otherwise, i.e. if $T'(y_j^-) = 0$, then it follows from the clause relations that $T'(y_j^+) = 1$ and thus $T(a) = 1$ or $T(b) = 1$. Therefore, the clause is satisfied. Hence, T is a satisfying truth assignment for ψ. This proves the claim.

Observe that ϕ has $2n+2m$ variables and $2n+3m$ relations. Moreover, ϕ can be constructed in polynomial time. Now apply the construction of Lemma 14.1 to ϕ in polynomial time. This yields an instance $(G_\phi, \mathcal{S}_\phi)$ of Planar Steiner Forest such that G_ϕ is a planar graph of treewidth 3, and $(G_\phi, \mathcal{S}_\phi)$ has a solution with at most $8n + 11m$ edges if and only if ϕ is satisfiable. Using the above claim, $(G_\phi, \mathcal{S}_\phi)$ has a solution with at most $8n + 11m$ edges if and only if ψ is satisfiable. Note that $8n + 11m \leq 35m$. Therefore, the existence of an algorithm as in the theorem statement would imply an algorithm that decides instances of 3-SAT in $2^{o(m)}$ time. This proves the theorem. □

Acknowledgements

We thank Daniel Lokshtanov and Saket Saurabh for showing us the application of Baker's approach to Planar Multiway Cut (Theorem 13.1) and for allowing us to include the proof in this paper. Moreover, we acknowledge the discussions with Daniel Lokshtanov that lead to the discovery that the NP-hardness proof for Steiner Forest on planar graphs of treewidth 3 of Bateni et al. [5] can be strengthened to also refute a subexponential-time algorithm.

We would like also to acknowledge the support and extremely productive atmosphere at Dagstuhl Seminars 13121, 13421 and 14071. At the first one, major technical ideas of the proof of Theorem 1.1 were developed. At the second one, the fundaments of the weighted variant (Theorem 1.7) were laid, whereas many important details were discussed and straightened during the third one.

References

[1] Ajit Agrawal, Philip N. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

[3] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.

[4] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Philip N. Klein, and Claire Mathieu. A polynomial-time approximation scheme for planar multiway cut. In Yuval Rabani, editor, *SODA*, pages 639–655. SIAM, 2012.

[5] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Dániel Marx. Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth. *J. ACM*, 58(5):21, 2011.

[6] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan sparsifiers. *SIAM J. Comput.*, 41(6):1704–1721, 2012.

[7] András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In Gary L. Miller, editor, *STOC*, pages 47–55. ACM, 1996.

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

[9] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[10] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. In *FOCS*, pages 629–638. IEEE Computer Society, 2009.

[11] Glencora Borradaile, Erik D. Demaine, and Siamak Tazari. Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs. *Algorithmica*, 68(2):287–311, 2014.

[12] Glencora Borradaile, Philip N. Klein, and Claire Mathieu. An $O(n \log n)$ approximation scheme for Steiner tree in planar graphs. *ACM Transactions on Algorithms*, 5(3), 2009.

[13] Karol Borsuk. Sur les rétractes. *Fundamenta Mathematicae*, 17:2–20, 1931.

[14] Vincent Bouchitté, Frédéric Mazoit, and Ioan Todinca. Treewidth of planar graphs: connections with duality. *Electronic Notes in Discrete Mathematics*, 10:34–38, 2001.

[15] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. An improved LP-based approximation for Steiner tree. In Leonard J. Schulman, editor, *STOC*, pages 583–592. ACM, 2010.

[16] Yixin Cao, Jianer Chen, and Jia-Hao Fan. An $O^*(1.84^k)$ parameterized algorithm for the multiterminal cut problem. In Leszek Gasieniec and Frank Wolter, editors, *FCT*, volume 8070 of *Lecture Notes in Computer Science*, pages 84–94. Springer, 2013.

[17] Jianer Chen, Yang Liu, and Songjian Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.

[18] Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as CNF-SAT. In *IEEE Conference on Computational Complexity*, pages 74–84. IEEE, 2012.

[19] Marek Cygan, Łukasz Kowalik, and Marcin Pilipczuk. Open problems from workshop on kernels, 2013. Available at `http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf`.

[20] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

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

[22] Erik D. Demaine and MohammadTaghi Hajiaghayi. Bidimensionality. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*. Springer, 2008.

[23] Erik D. Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *Comput. J.*, 51(3):292–302, 2008.

[24] Erik D. Demaine and MohammadTaghi Hajiaghayi. Linearity of grid minors in treewidth with applications through bidimensionality. *Combinatorica*, 28(1):19–36, 2008.

[25] Reinhard Diestel. *Graph Theory*. Springer, 2005.

[26] Krzysztof Diks and Piotr Sankowski. Dynamic plane transitive closure. In Lars Arge, Michael Hoffmann, and Emo Welzl, editors, *ESA*, volume 4698 of *Lecture Notes in Computer Science*, pages 594–604. Springer, 2007.

[27] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and IDs. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP (1)*, volume 5555 of *LNCS*, pages 378–389. Springer, 2009.

[28] Frederic Dorn, Fedor V. Fomin, and Dimitrios M. Thilikos. Catalan structures and dynamic programming in H-minor-free graphs. *J. Comput. Syst. Sci.*, 78(5):1606–1622, 2012.

[29] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer, 1999.

[30] Andrew Drucker. New limits to classical and quantum instance compression. In *FOCS*, pages 609–618. IEEE Computer Society, 2012.

[31] David Eisenstat and Philip N. Klein. Linear-time algorithms for max flow and multiple-source shortest paths in unit-weight planar graphs. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC*, pages 735–744. ACM, 2013.

[32] David Eisenstat, Philip N. Klein, and Claire Mathieu. An efficient polynomial-time approximation scheme for steiner forest in planar graphs. In Yuval Rabani, editor, *SODA*, pages 626–638. SIAM, 2012.

[33] David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Amnon Nissenzweig. Sparsification — a technique for speeding up dynamic graph algorithms. *J. ACM*, 44(5):669–696, September 1997.

[34] David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Thomas H. Spencer. Separator based sparsification: I. planarity testing and minimum spanning trees. *Journal of Computer and System Sciences*, 52(1):3 – 27, 1996.

[35] David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Thomas H. Spencer. Separator-based sparsification ii: Edge and vertex connectivity. *SIAM J. Comput.*, 28(1):341–381, 1998.

[36] Ranel E. Erickson, Clyde L. Monma, and Arthur F. Jr. Veinott. Send-and-split method for minimum-concave-cost network flows. *Mathematics of Operations Research*, 12(4):634–664, 1987.

[37] Michael R. Fellows, Jiong Guo, Dániel Marx, and Saket Saurabh. Data reduction and problem kernels (dagstuhl seminar 12241). *Dagstuhl Reports*, 2(6):26–50, 2012.

[38] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms.* Texts in Theoretical Computer Science. Springer, 2010.

[39] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In Moses Charikar, editor, *SODA*, pages 503–510. SIAM, 2010.

[40] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[41] Michael R. Garey and David S. Johnson. The Rectilinear Steiner Tree problem is NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.

[42] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *J. Soc. Indust. Appl. Math.*, 9(4):551–570, 1961.

[43] Torben Hagerup, Jyrki Katajainen, Naomi Nishimura, and Prabhakar Ragde. Characterizing multiterminal flow networks and computing flows in networks of small treewidth. *J. Comput. Syst. Sci.*, 57(3):366–375, 1998.

[44] Monika Rauch Henzinger and Valerie King. Randomized fully dynamic graph algorithms with polylogarithmic time per operation. *J. ACM*, 46(4):502–516, 1999.

[45] Monika Rauch Henzinger, Philip N. Klein, Satish Rao, and Sairam Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.*, 55(1):3–23, 1997.

[46] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[47] Giuseppe F. Italiano, Yahav Nussbaum, Piotr Sankowski, and Christian Wulff-Nilsen. Improved algorithms for min cut and max flow in undirected planar graphs. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 313–322. ACM, 2011.

[48] David R. Karger, Philip N. Klein, Clifford Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.

[49] Richard Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[50] Philip N. Klein. Multiple-source shortest paths in planar graphs. In *SODA*, pages 146–155. SIAM, 2005.

[51] Philip N. Klein. A subset spanner for planar graphs, with application to Subset TSP. In Jon M. Kleinberg, editor, *STOC*, pages 749–756. ACM, 2006.

[52] Philip N. Klein and Dániel Marx. Solving planar k-terminal cut in $O(n^{c\sqrt{k}})$ time. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 569–580. Springer, 2012.

[53] Philip N. Klein and Sairam Subramanian. A fully dynamic approximation scheme for shortest paths in planar graphs. *Algorithmica*, 22(3):235–249, 1998.

[54] Stefan Kratsch and Magnus Wahlström. Compression via matroids: a randomized polynomial kernel for odd cycle transversal. In Yuval Rabani, editor, *SODA*, pages 94–103. SIAM, 2012.

[55] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *FOCS*, pages 450–459. IEEE Computer Society, 2012.

[56] Robert Krauthgamer and Inbal Rika. Mimicking networks and succinct representations of terminal cuts. In Sanjeev Khanna, editor, *SODA*, pages 1789–1799. SIAM, 2013.

[57] Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.

[58] Dániel Marx. A tight lower bound for planar multiway cut with fixed number of terminals. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 677–688. Springer, 2012.

[59] Hiroshi Nagamochi and Toshihide Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5&6):583–596, 1992.

[60] Jesper Nederlof. Fast polynomial-space algorithms using Möbius inversion: Improving on Steiner Tree and related problems. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *ICALP (1)*, volume 5555 of *LNCS*, pages 713–725. Springer, 2009.

[61] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications.* Oxford University Press, 2006.

[62] David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

[63] Marcin Pilipczuk, Michal Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Subexponential-time parameterized algorithm for steiner tree on planar graphs. In Natacha Portier and Thomas Wilke, editors, *STACS*, volume 20 of *LIPIcs*, pages 353–364. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[64] Heike Ripphausen-Lipa, Dorothea Wagner, and Karsten Weihe. The vertex-disjoint menger problem in planar graphs. *SIAM J. Comput.*, 26(2):331–349, 1997.

[65] Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Comb. Theory, Ser. B*, 62(2):323–348, 1994.

[66] Sairam Subramanian. A fully dynamic data structure for reachability in planar digraphs. In Thomas Lengauer, editor, *ESA*, volume 726 of *Lecture Notes in Computer Science*, pages 372–383. Springer, 1993.

[67] Siamak Tazari. Faster approximation schemes and parameterized algorithms on H-minor-free and odd-minor-free graphs. In Petr Hlinený and Antonín Kucera, editors, *MFCS*, volume 6281 of *LNCS*, pages 641–652. Springer-Verlag, Berlin, 2010.