

Multicut Problems in Almost-Planar Graphs: the Dependency of Complexity on the Demand Pattern

Florian Hörsch 

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Dániel Marx 

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Abstract

Given a graph G , a set T of terminal vertices, and a demand graph H on T , the MULTICUT problem asks for a set of edges of minimum weight that separates the pairs of terminals specified by the edges of H . The MULTICUT problem can be solved in polynomial time if the number of terminals and the genus of the graph is bounded (Colin de Verdière [Algorithmica, 2017]). Restricting the possible demand graphs in the input leads to special cases of MULTICUT whose complexity might be different from the general problem.

Focke et al. [SoCG 2024] systematically characterized which special cases of MULTICUT are fixed-parameter tractable parameterized by the number of terminals on planar graphs. Moreover, extending these results beyond planar graphs, they precisely determined how the parameter genus influences the complexity and presented partial results of this form for graphs that can be made planar by the deletion of π edges. Continuing this line of work, we complete the picture on how this parameter π influences the complexity of different special cases and precisely determine the influence of the crossing number, another parameter measuring closeness to planarity.

Formally, let \mathcal{H} be any class of graphs (satisfying a mild closure property) and let $\text{MULTICUT}(\mathcal{H})$ be the special case when the demand graph H is in \mathcal{H} . Our first main result is showing that if \mathcal{H} has the combinatorial property of having *bounded distance to extended bicliques*, then $\text{MULTICUT}(\mathcal{H})$ on *unweighted* graphs is FPT parameterized by the number t of terminals and π . For the case when \mathcal{H} does not have this combinatorial property, Focke et al. [SoCG 2024] showed that $O(\sqrt{t})$ is essentially the best possible exponent of the running time; together with our result, this gives a complete understanding of how the parameter π influences complexity on unweighted graphs. Our second main result is giving an algorithm whose existence shows that the parameter crossing number behaves analogously if we consider $\text{MULTICUT}(\mathcal{H})$ on *weighted* graphs.

2012 ACM Subject Classification Mathematics of computing → Graphs and surfaces; Mathematics of computing → Graph algorithms; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases MultiCut, Multiway Cut, Parameterized Complexity, Tight Bounds, Embedded Graph, Planar Graph, Crossing Number

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.87

Related Version *Full Version:* <https://arxiv.org/abs/2504.21624> [9]

Acknowledgements We want to thank Jacob Focke and Shaohua Li for the helpful discussions.

1 Introduction

Most NP-hard problems remain NP-hard when restricted to planar graphs, unless they become trivial or irrelevant on planar graphs. There are very few NP-hard problems that are polynomial-time solvable on planar graphs for some nontrivial and interesting reason. One such case is the MULTIWAY CUT problem: given a graph G and a set $T \subseteq V(G)$ of *terminals*, the task is to find a *multiway cut* S of minimum weight, that is, a set $S \subseteq E(G)$



© Florian Hörsch and Dániel Marx;
licensed under Creative Commons License CC-BY 4.0
33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 87; pp. 87:1–87:13



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

such that every component of $G \setminus S$ contains at most one terminal. Dahlhaus et al. [3] showed that MULTIWAY CUT in general graphs is NP-hard even for three terminals, while it is polynomial-time solvable on planar graphs for every fixed number $|T|$ of terminals. The exponent of n in the algorithm of Dahlhaus et al. [3] is $O(|T|)$. Later, the dependence was improved to $O(\sqrt{|T|})$ [2, 10], which was shown to be optimal under the Exponential-Time Hypothesis (ETH) [1, 11]. The results were extended to bounded-genus graphs, again with an essentially best possible form of the exponent [1, 2].

MULTICUT is a generalization of MULTIWAY CUT where not all terminals need to be separated from each other, but the input specifies which pairs of terminals need to be separated (and we do not care whether other pairs are separated or not).

MULTICUT

Input: A weighted graph G together with a set $T \subseteq V(G)$ of *terminals*;
and a *demand graph* H with $V(H) = T$.

Output: A minimum-weight set $S \subseteq E(G)$ such that u and v are in distinct components of $G \setminus S$ whenever uv is an edge of H .

Let us observe that MULTIWAY CUT is the special case of MULTICUT when H is a complete graph on T . Colin de Verdière [2] presented an algorithm for MULTICUT where the exponent of the running time is $O(\sqrt{|T|})$, similarly to MULTIWAY CUT. However, it is no longer clear anymore that this algorithm is a tight, optimal result and cannot be improved in some cases: it could be that there are special cases, special type of demand graphs where significantly better algorithms are possible. For example, if H is a biclique (complete bipartite graph), then MULTICUT can be solved in polynomial time by a reduction to a minimum weight $s - t$ cut problem. A very natural question to consider is the case when H is a complete 3-partite graph. This special case is the GROUP 3-TERMINAL CUT problem: given a graph G with three sets of terminals T_1, T_2, T_3 , the task is to find a set S of edges of minimum total weight such that $G \setminus S$ has no path between any vertex of T_i and any vertex of T_j for all distinct $i, j \in [3]$. While a general MULTICUT algorithm solves GROUP 3-TERMINAL CUT with the exponent of the running time being the square root of the number of terminals, the lower bounds for MULTIWAY CUT do not imply that this dependence is optimal.

Focke et al. [5] systematically analyzed which types of demand graphs make the problem more tractable. The question was explored in the following setting. Let \mathcal{H} be a class of graphs (i.e., a set of graphs closed under isomorphism). Then we denote by $\text{MULTICUT}(\mathcal{H})$ the special case of MULTICUT that contains only instances (G, H) where $H \in \mathcal{H}$. For example, if \mathcal{H} is the class of cliques, then $\text{MULTICUT}(\mathcal{H})$ is exactly MULTIWAY CUT.

Ideally, for every class \mathcal{H} of graphs, we would like to understand the best possible running time of $\text{MULTICUT}(\mathcal{H})$ on planar graphs. In order to make the question more robust, Focke et al. [5] considered only classes \mathcal{H} satisfying a mild closure property. We say that H' is a *projection* of H if H' can be obtained from H by deleting vertices and identifying pairs of independent vertices. Moreover, a class of graphs \mathcal{H} is *projection-closed* if, whenever $H \in \mathcal{H}$ and H' is a projection of H , then $H' \in \mathcal{H}$ also holds. The intuition is that an instance having demand graph H' can be easily simulated by an instance having demand graph H , thus it is reasonable to assume we only consider classes of demand graphs closed under this operation.

Let us consider some easy cases first. As we mentioned before, the MULTICUT problem can be solved in polynomial time by a reduction to $s - t$ minimum cut if the demand graph is a biclique. Furthermore, if H has only two edges, then a similar reduction is known [8, 12]. Isolated vertices of H clearly do not play any role in the problem. We say that H is a *trivial pattern* if, after removing isolated vertices, it is either a biclique or has at most two edges.

An *extended biclique* is a graph that consists of a complete bipartite graph together with a set of isolated vertices. Let μ be the minimum number of vertices that need to be deleted to make a graph H an extended biclique. Then we say that H has distance μ to extended bicliques. Moreover, we say that a graph class \mathcal{H} has *bounded distance to extended bicliques* if there is a constant μ such that every $H \in \mathcal{H}$ has distance at most μ to extended bicliques.

For planar graphs, Focke et al. [5] give the following characterization:

► **Theorem 1.1** ([5]). *Let \mathcal{H} be a computable projection-closed class of graphs. Then the following holds for $\text{MULTICUT}(\mathcal{H})$ on planar graphs.*

1. *If \mathcal{H} has bounded distance to extended bicliques, then there is an $f(t)n^{O(1)}$ time algorithm.*
2. *Otherwise,*
 - a. *There is an $f(t)n^{O(\sqrt{t})}$ time algorithm.*
 - b. *Assuming ETH, there is a universal constant $\alpha > 0$ such that for any fixed choice of $t \geq 3$, there is no $O(n^{\alpha\sqrt{t}})$ algorithm, even when restricted to unweighted instances with at most t terminals.*

That is, depending on whether \mathcal{H} has bounded distance to extended bicliques, the number t of terminals has to appear in the exponent. Focke et al. [5] precisely characterized how the complexity changes if we move from planar graphs to bounded-genus graphs. As we can see, the parameter genus can appear in the exponent in different ways, depending on \mathcal{H} .

► **Theorem 1.2** ([5]). *Let \mathcal{H} be a computable projection-closed class of graphs. Then the following holds for $\text{MULTICUT}(\mathcal{H})$.*

1. *If every graph in \mathcal{H} is a trivial pattern, then there is a polynomial time algorithm.*
2. *Otherwise, if \mathcal{H} has bounded distance to extended bicliques, then*
 - a. *There is an $f(g, t)n^{O(g)}$ time algorithm.*
 - b. *Assuming ETH, there is a universal constant $\alpha > 0$ such that for any fixed choice of $\vec{g} \geq 0$, there is no $O(n^{\alpha(\vec{g}+1)/\log(\vec{g}+2)})$ algorithm, even when restricted to unweighted instances with orientable genus at most \vec{g} and $t = 3$ terminals.*
3. *Otherwise,*
 - a. *There is an $f(g, t)n^{O(\sqrt{g^2+gt+t})}$ time algorithm.*
 - b. *Assuming ETH, there is a universal constant $\alpha > 0$ such that for any fixed choice of $\vec{g} \geq 0$ and $t \geq 3$, there is no $O(n^{\alpha\sqrt{\vec{g}^2+\vec{g}t+t}/\log(\vec{g}+t)})$ algorithm, even when restricted to unweighted instances with orientable genus at most \vec{g} and at most t terminals.*

Focke et al. [5] also considered a much more restricted extension of planar graphs than bounded-genus graphs: graphs that can be made planar by deleting a few edges. We use $\text{planar} + \pi e$ for the class of graphs that can be made planar by removing at most π edges. Given an instance (G, H) , we use π for the smallest integer such that G is in $\text{planar} + \pi e$.

Observe that in Theorem 1.2, the algorithmic results are stated for the edge-weighted version, while the lower bounds are stated for the unweighted version. In fact, edge-weights do not make much difference in Theorem 1.2, as polynomial edge weights can be simulated by parallel edges without changing the genus of the graphs. However, this simulation can easily change the number π of edges that needs to be deleted to make the graph planar. This means that if we consider the influence of the parameter π on the running time, the edge-weighted and the unweighted problem may behave differently. Focke et al. [5] characterized the influence of this parameter in the edge-weighted case:

► **Theorem 1.3** ([5]). *Let \mathcal{H} be a computable projection-closed class of graphs. Then the following holds for edge-weighted $\text{MULTICUT}(\mathcal{H})$.*

1. *If every graph in \mathcal{H} is a trivial pattern, then there is a polynomial-time algorithm.*
2. *Otherwise, if \mathcal{H} has bounded distance to extended bicliques, then*
 - a. *There is an $f(\pi, t)n^{O(\sqrt{\pi})}$ time algorithm.*
 - b. *Assuming ETH, there is a universal constant $\alpha > 0$ such that for any fixed choice of $\pi \geq 0$, there is no $O(n^{\alpha\sqrt{\pi}})$ algorithm, even when restricted to instances in $\text{planar} + \pi e$ with $t = 3$ terminals.*
3. *Otherwise,*
 - a. *There is an $f(\pi, t)n^{O(\sqrt{\pi+t})}$ algorithm.*
 - b. *Assuming ETH, there is a universal constant $\alpha > 0$ such that for any fixed choice of $\pi \geq 0$ and $t \geq 3$, there is no $O(n^{\alpha\sqrt{\pi+t}})$ algorithm, even when restricted to $\text{planar} + \pi e$ instances with at most t terminals.*

Observe that π has a different form of influence on the exponent compared to genus: it is $O(\sqrt{\pi})$ instead of $O(g)$ in the case of bounded distance to extended bicliques, and $O(\sqrt{\pi+t})$ instead of $O(\sqrt{g^2+gt+t})$ in the unbounded distance case. Intuitively, having π extra edges is a much more restricted extension of planar graphs than having genus g , and consequently, the parameter π has a much weaker influence on the exponent than genus has.

For unweighted graphs, Focke et al. [5] showed that the problem is easier than the edge-weighted version and the parameter π can be removed from the exponent.

► **Theorem 1.4** ([5]). *Unweighted MULTICUT can be solved in time $f(\pi, t)n^{O(\sqrt{t})}$.*

However, this result leaves open the question what happens in unweighted $\text{MULTICUT}(\mathcal{H})$ if \mathcal{H} has bounded distance to extended bicliques. Based on Theorems 1.2 and 1.3, one would expect that it is possible to remove the number t of terminals from the exponent as well.

Our contributions

First, we resolve the remaining question about unweighted $\text{MULTICUT}(\mathcal{H})$ and show that if \mathcal{H} has bounded distance to extended bicliques, then both t and π can be removed from the exponent. Recall that μ is the number of vertices needed to be deleted to make the demand graph an extended biclique. Our first main result is the following algorithm.

► **Theorem 1.5.** *Unweighted MULTICUT can be solved in time $f(\pi, t)n^{O(\sqrt{\mu})}$.*

Observe that, in contrast to Theorem 1.4, only μ is in the exponent of the running time of the algorithm of Theorem 1.5. As $\mu \leq t$, this algorithm gives an independent proof of Theorem 1.4 with a very different proof technique. With Theorem 1.5 at hand, we can complete the picture for the complexity of unweighted $\text{MULTICUT}(\mathcal{H})$:

► **Theorem 1.6.** *Let \mathcal{H} be a computable projection-closed class of graphs. Then the following holds for unweighted $\text{MULTICUT}(\mathcal{H})$.*

1. *If \mathcal{H} has bounded distance to extended bicliques, then there is an $f(\pi, t)n^{O(1)}$ -time algorithm.*
2. *Otherwise,*
 - a. *There is an $f(\pi, t)n^{O(\sqrt{t})}$ algorithm.*
 - b. *Assuming ETH, there is a universal constant $\alpha > 0$ such that for any fixed choice of $t \geq 3$, there is no $O(n^{\alpha\sqrt{t}})$ -time algorithm, even when restricted to planar instances with at most t terminals.*

We consider yet another, even more restricted parameter measuring the distance from planar graphs: the *crossing number*, the minimum number of crossings needed when drawing the graph in the plane. Clearly, if a graph has a drawing with cr crossings, then we can remove a set of at most cr edges to make the graph planar. Thus the crossing number cr is always at least π , that is, an algorithm parameterized by π immediately gives an algorithm parameterized by cr .

Again, there is no straightforward reduction from the edge-weighted version to the unweighted version without increasing the crossing number. Thus both versions of the problem when parameterizing by crossing number are interesting. Clearly, the unweighted version is no harder than the edge-weighted one. Moreover, both versions parameterized by crossing number are no harder than their respective versions parameterized by π . But which of these versions are *strictly* harder than the others?

It turns out that, when parameterizing by crossing number, the edge weights do not change the hardness of the problem and both the edge-weighted and the unweighted versions parameterized by cr have the same complexity as the *unweighted* version parameterized by π . The following algorithm is our main result on the crossing number parameterization.

► **Theorem 1.7.** *Edge-weighted MULTICUT can be solved in time $f(cr, t)n^{O(\sqrt{\mu})}$.*

Together with Theorem 1.1, it implies the following complete classification for the complexity of $\text{MULTICUT}(\mathcal{H})$ when parameterizing by cr :

► **Theorem 1.8.** *Let \mathcal{H} be a computable projection-closed class of graphs. Then the following holds for edge-weighted $\text{MULTICUT}(\mathcal{H})$.*

1. *If \mathcal{H} has bounded distance to extended bicliques, then there is an $f(cr, t)n^{O(1)}$ -time algorithm.*
2. *Otherwise,*
 - a. *There is an $f(cr, t)n^{O(\sqrt{t})}$ algorithm.*
 - b. *Assuming ETH, there is a universal constant $\alpha > 0$ such that for any fixed choice of $t \geq 3$, there is no $O(n^{\alpha\sqrt{t}})$ -time algorithm, even when restricted to unweighted planar instances with at most t terminals.*

Note that the algorithmic results of Theorems 1.6 and 1.8 are incomparable: Theorem 1.6 considers only unweighted graphs, while Theorem 1.8 considers edge-weighted graphs, but under parameterization by a potentially much larger parameter.

Theorems 1.2–1.8 complete the picture of how different parameters measuring closeness to planarity influence the running time of $\text{MULTICUT}(\mathcal{H})$ for different classes \mathcal{H} (see Figure 1). Let us observe that the exact form of the influence is very different for the different parameters. As we explain below, the treewidth of the so-called multicut dual of the optimum solution is the main factor determining the exponent of the running time [2, 5]. The exponent $O(\sqrt{t})$ for planar graphs comes from the fact that a planar graph with $O(t)$ vertices (or faces) has treewidth $O(\sqrt{t})$. Intuitively, for nonplanar graphs, the situation changes as follows:

- **Genus.** If the genus is large, then a graph with t vertices can have treewidth larger than $O(\sqrt{t})$. The maximum possible treewidth is given by the (nonobvious) formula $O(\sqrt{g^2 + gt + t})$. Therefore, the maximum treewidth of the multicut dual is $O(\sqrt{g^2 + gt + t})$, giving a bound on the exponent of the running time.
- **Edge-weighted planar + πe .** With extra edges having infinite (or very large) weight, we can simulate multiple terminals at different parts of a planar graph. More precisely, a planar instance of GROUP 3-TERMINAL CUT with t terminals in each of the three groups can be reduced to an instance of MULTIWAY CUT with three terminals if we connect

	genus	weighted planar + πe	unweighted planar + πe	crossing number
\mathcal{H} with bounded distance to extended bicliques	$f(g, t)n^{O(g)}$	$f(\pi, t)n^{O(\sqrt{\pi})}$	$f(\pi, t)n^{O(1)} (*)$	$f(cr, t)n^{O(1)} (*)$
\mathcal{H} with unbounded distance to extended bicliques	$f(g, t)n^{O(\sqrt{g^2+gt+t})}$	$f(\pi, t)n^{O(\sqrt{\pi+t})}$	$f(\pi, t)n^{O(\sqrt{t})}$	$f(cr, t)n^{O(\sqrt{t})} (*)$

■ **Figure 1** The running time of the algorithms for $\text{MULTICUT}(\mathcal{H})$ with the different parameterizations. The new results of the paper are marked by (*). Note that, for every fixed projection-closed class \mathcal{H} , the exponents of the running time are tight (up to logarithmic factors).

the terminals in each group with a total of $\pi = 3(t - 1)$ infinite-weight edges, creating a **planar + πe** instance. Thus these extra edges influence the running time exactly as extra terminals would do.

- **Unweighted planar + πe .** We can always try to solve the problem by putting the π extra edges into the solution, removing them from G , and then solving the resulting planar instance. Of course, this might not always lead to an optimum solution, but can be up to π edges larger. However, this argument allows us to say that if the solution contains a cut that is “inefficient” in the sense that it is larger than the minimum cut by more than π , then the cut can be replaced by a minimum cut and the π extra edges, resulting in a strictly smaller solution. Using these type of arguments, the problem can be reduced to a planar setting where the number of faces of the multicut dual does not depend on π .
- **Crossing number.** Each crossing can increase the number of vertices of the multicut dual by one, thus the number of vertices and the treewidth of the multicut dual can be $O(t + cr)$ and $O(\sqrt{t} + cr)$, respectively. However, all these extra vertices of the multicut dual are at fixed places determined by the crossing. As pointed out by Focke et al. [6] it is sufficient to consider the treewidth of the multicut dual *after removing all the vertices whose locations are fixed* (see Theorem 2.3). Indeed, after removing these vertices, the treewidth of the multicut dual does not depend on the crossing number.

Some remarks on two further notion of almost-planar graphs can be found in the full version.

2 Preliminaries

Some basic notation on sets, graphs, and multicuts can be found in the full version. We now give some less well-known, problem-specific definitions.

An *extended biclique decomposition* of a graph H is a partition (B_1, B_2, I, X) of $V(H)$ such that all vertices of I are isolated in $H - X$, $E(H)$ contains the edge $t_1 t_2$ for all $t_1 \in B_1$ and $t_2 \in B_2$, and $H[B_i]$ is an edgeless graph for $i \in [2]$. We use \mathcal{H}_μ for the set of graphs H that admit an extended biclique decomposition (B_1, B_2, I, X) with $|X| \leq \mu$.

We now review the concept of so-called multicut duals, which will play a crucial role throughout the article. Recall that an instance (G, H) of MULTICUT is *planar* if G is given in the form of a graph cellularly embedded in the plane. Given a graph C that is embedded in the plane and in general position with G , we denote by $e_G(C)$ those edges of G that are crossed by at least one edge of C . The *weight* $w(C)$ of C is defined to be $w(e_G(C))$.

where w is the weight function associated with G . A *multicut dual* for (G, H) is a graph C embedded in the plane that is in general position with G and has the property that, for every $t_1, t_2 \in V(H)$ with $t_1 t_2 \in E(H)$, the terminals t_1 and t_2 are contained in different faces of C .

The following result is an immediate consequence of Lemmas 2.3 and 2.4 in [6].

► **Lemma 2.1.** *Let (G, H) be a planar instance of MULTICUT and C a minimum weight multicut dual for (G, H) . Then $e_G(C)$ is a minimum weight multicut for (G, H) .*

In order to make use of the topology of certain multicut duals, we need two results related to treewidth. The first one of the following two results provides a sufficient criterion for the treewidth of certain graphs being bounded and the second one allows to exploit bounded treewidth algorithmically.

The following result can be obtained from Theorem 3.2 in [4] and the fact that the treewidth of any graph can only be larger than its branchwidth by a constant factor.

► **Proposition 2.2.** *Let C be a planar graph that admits a 5-dominating set of size k for some positive integer k . Then $\text{tw}(C) = O(\sqrt{k})$.*

An *extended* planar instance (G, H, F^*) of MULTICUT is a planar instance (G, H) of MULTICUT together with a set F^* of faces of G which is given with the input. Given a set F^* of faces of G and a multicut dual C for (G, H) , let $C - F^*$ be the graph obtained from C by removing every vertex that is in a face of F^* and removing every edge that intersects a face of F^* . The following result is the restriction of Theorem 1.11 in [6] to the planar setting.

► **Theorem 2.3.** *Let (G, H, F^*) be an extended planar instance of MULTICUT such that every subcubic inclusionwise minimal minimum-weight multicut dual C satisfies $\text{tw}(C - F^*) \leq \beta$. Then an optimum solution of (G, H) can be found in time $f(|F^*|, t)n^{O(\beta)}$.*

3 Edge-deletion distance to planarity

In this section, we give an overview of the proof of the following result, whose complete proof can be found in the full version.

► **Theorem 1.5.** *Unweighted MULTICUT can be solved in time $f(\pi, t)n^{O(\sqrt{\pi})}$.*

Throughout this section, we say that an instance (G, H) of MULTICUT is **planar** + πe if a set $E_\pi \subseteq E(G)$ is given with the instance such that $|E_\pi| = \pi$ and $G \setminus E_\pi$ is planar. We further use T for $V(H)$, W for $V(E_\pi)$, and G_0 for $G \setminus E_\pi$. Also, for some $S \subseteq E(G)$, we use S_0 for $S \setminus E_\pi$. We say that (G, H) is *connected* if G is connected.

The overall algorithmic strategy for Theorem 1.5 is to run an algorithm on the given instance that either produces a minimum multicut for the instance or computes a collection of instances for which the parameters in consideration are smaller and such that from minimum multicuts of all those instances, a minimum multicut for the original instance can be computed. We need the following definitions.

Given instances (G, H) and (G', H') of MULTICUT, (G', H') is a *subinstance* of (G, H) if G' is a labelled subgraph of G and H' is a labelled induced subgraph of H . Next, an *extended subinstance* of (G, H) consists of a set $S' \subseteq E(G)$ and a subinstance (G', H') of (G, H) with $E(G') \cap S' = \emptyset$. When we say that an extended subinstance $(S', (G', H'))$ has certain properties which can only be associated to an instance of MULTICUT, we refer to (G', H') . Next, we say that an extended subinstance $(S', (G', H'))$ of (G, H) is *optimumbound* for (G, H) if $S' \cup S''$ is a minimum multicut for (G, H) for every minimum multicut S'' for (G', H') .

The main technical difficulty is to prove the following result, which shows that the above described reduction to smaller instances exists for connected instances. With this result at hand, Theorem 1.5 follows readily. In particular, if disconnected instances are created, they can easily be reduced to instances corresponding to the components of the graph. Further observe that μ cannot increase when moving to a subinstance as the new demand graph is an induced subgraph of the previous one.

► **Lemma 3.1.** *Let (G, H) be a connected planar + πe instance of MULTICUT. Then, in $f(\pi, t)n^{O(\sqrt{\mu})}$, we can run an algorithm that returns a set $S \subseteq E(G)$ and a collection $(S_i, (G_i, H_i))_{i \in [k]}$ of extended subinstances of (G, H) for some $k = f(\pi, t)$ such that for all $i \in [k]$, we have that $\pi(G_i) + |V(H_i)| < \pi(G) + |V(H)|$ holds and either S is a minimum multicut for (G, H) or there exists some $i \in [k]$ such that $(S_i, (G_i, H_i))$ is optimumbound for (G, H) .*

We now give an overview of the proof of Lemma 3.1. Let (G, H) be a planar + πe instance of MULTICUT. The idea is to have a branching algorithm that gradually guesses more and more crucial information about a minimum multicut for (G', H') while the number of directions it branches into remains bounded. The current status of this procedure is displayed by a subpartition of the vertices of $V(G)$ with some extra properties. More formally, a *state* is a subpartition \mathcal{P} of $V(G)$ with $W \cup T \subseteq \bigcup \mathcal{P}$ and $Y \cap (W \cup T) \neq \emptyset$ for all $Y \in \mathcal{P}$. Given a certain state, the objective is to understand whether there exists a minimum multicut S for (G, H) such that the structure of the components of $G_0 \setminus S$ reflects the structure of \mathcal{P} . In order to make this more formal, we need the following definitions: Given a set V and two subpartitions \mathcal{P}^1 and \mathcal{P}^2 of V , we say that \mathcal{P}^2 *extends* \mathcal{P}^1 if $|\mathcal{P}^1| = |\mathcal{P}^2|$ and for every $Y \in \mathcal{P}^1$, there exists some $\bar{Y} \in \mathcal{P}^2$ with $Y \subseteq \bar{Y}$ such that for any distinct $Y_1, Y_2 \in \mathcal{P}^1$, we have $\bar{Y}_1 \neq \bar{Y}_2$. Given a set $S \subseteq E(G)$, we use S_0 for $S \cap E(G_0)$ and $\mathcal{K}(G_0 \setminus S_0)$ for the unique partition of $V(G)$ in which two vertices are in the same partition class if and only if they are in the same component of $G_0 \setminus S_0$. For a state \mathcal{P} , we say that a multicut S for (G, H) *respects* \mathcal{P} if $\mathcal{K}(G_0 \setminus S_0)$ extends \mathcal{P} and we say that a state is *valid* if there exists a minimum multicut for (G, H) respecting it. Further, \mathcal{P} is *maximum valid* if $|\mathcal{P}|$ is maximum among all valid states.

The idea of our algorithm now consists of considering a valid state, choosing a vertex that is not contained in any class of the state and making the classes larger by adding this vertex to one of them. As we do not know which class this vertex should be added to, we try all classes of the state and branch in these directions. The difficulty is now to carefully choose the vertex in consideration and make some additional modifications on the states so that this procedure terminates sufficiently fast.

In order to be able to explain how this works, we first need some more definitions. Given disjoint sets $Y_1, Y_2 \subseteq V(G)$, we use $\lambda_{G_0}(Y_1, Y_2)$ to denote the size of the smallest set of edges in $E(G_0)$ whose deletion from G_0 results in a graph in which there exists no component containing both a vertex of Y_1 and a vertex of Y_2 . Next, for a state \mathcal{P} and some $Y \in \mathcal{P}$, we define $\alpha_{G, \mathcal{P}}(Y) = \lambda_{G_0}(Y, \bigcup \mathcal{P} \setminus Y) - \lambda_{G_0}(Y \cap (W \cup T), W \cup T \setminus Y)$. Intuitively speaking, this measure describes how much more costly it is to separate the class from the other classes of the state in comparison to only separating the corresponding elements of $W \cup T$. Given a state \mathcal{P} , we now classify its classes according to their value of α and the value of α of their neighboring classes. More formally, given a state \mathcal{P} and some $Y \in \mathcal{P}$, we say that Y is *fat* in \mathcal{P} if $\alpha_{G_0, \mathcal{P}}(Y) \geq \pi(t+1) + 1$. For some $Y \in \mathcal{P}$ that is not fat, we say that Y is *fat-neighboring* in \mathcal{P} if there exists some fat $Y' \in \mathcal{P}$, some $u \in Y$, and some $v \in Y'$ such that $uv \in E(G_0)$. Finally, for some $Y \in \mathcal{P}$ which is neither fat nor fat-neighboring in \mathcal{P} , we say that Y is *thin* in \mathcal{P} . We refer by \mathcal{P}_{fat} , $\mathcal{P}_{\text{f-n}}$, and $\mathcal{P}_{\text{thin}}$ to the fat, fat-neighboring, and thin classes of \mathcal{P} , respectively.

In order to make use of this distinction, we crucially need a structural property of minimum multicuts for (G, H) . Namely, the following result shows that, for any minimum multicut S for (G, H) and any vertex set of a component of $G_0 \setminus S_0$, one of two things holds: either the value of α is not too large or a terminal of X is close to this vertex set in a certain sense, where (B_1, B_2, I, X) is an extended biclique decomposition of H . In order to measure this proximity of certain vertex sets with respect to a multicut, we introduce the following notion: Let $S \subseteq E(G)$, and $Y_1, Y_2 \subseteq V(G)$. Then we denote by $\text{dist}_{G_0, S_0}(Y_1, Y_2)$ the smallest integer q such that G_0 contains a $Y_1 Y_2$ -path P with $|E(P) \cap S_0| = q$. We are now ready to state the following structural result for minimum multicuts.

► **Lemma 3.2.** *Let (G, H) be a planar + πe instance of MULTICUT, let S be a minimum multicut for (G, H) and let $Y \in \mathcal{P}_{\text{fat}}$ for $\mathcal{P} = \mathcal{K}(G_0 \setminus S_0)$. Then $\text{dist}_{G_0, S_0}(Y, X) \leq 1$.*

We now explain the strategy of our algorithm. Given a state with an extra property that will be explained later, we choose a vertex that is not contained in any class of the state and has a neighbor in a thin class of the state in G_0 . We then guess a partition class which this vertex will be added to and modify the state accordingly. In order to show the functionality of this algorithm, there are two points that need further explanation. First, we need to explain how to conclude when the described modification is no longer possible and second, we need to explain why this procedure terminates sufficiently fast.

We first explain the scenario that we can no longer execute the described modification, so there does not exist any vertex in the graph that is not contained in any class of the state \mathcal{P} in consideration and that has a neighbor in a thin class of \mathcal{P} in G_0 . We call such a state *complete*, all other states are called *incomplete*. For complete states, we need to consider two different cases, depending on whether \mathcal{P} contains a thin class.

The case that there exists a thin class Y can be handled with elementary means. Observe that for this thin class, all its neighbors in G_0 are contained in a different class of \mathcal{P} . Hence, if \mathcal{P} is valid, then all edges in $\delta_{G_0}(Y)$ are contained in a minimum multicut for (G, H) . We can hence delete these edges and solve the remaining instance, which turns out to be significantly smaller in a certain sense. This is subsumed in the following result.

► **Lemma 3.3.** *Let (G, H) be a connected planar + πe instance of MULTICUT and let a complete valid state \mathcal{P} of (G, H) with $\mathcal{P}_{\text{thin}} \neq \emptyset$ be given. Then in polynomial time, we can compute an optimum bound extended subinstance $(S', (G', H'))$ of (G, H) such that $\pi(G') + |V(H')| < \pi(G) + |V(H)|$.*

We now turn to the case that \mathcal{P} does not contain a thin set. In order to handle this case, geometrical arguments on multicut duals will be needed. Namely, we show that if \mathcal{P} is valid and does not contain a thin set, then the problem of finding a minimum multicut for (G, H) can be reduced to solving a planar instance of multicut, which can be solved sufficiently fast due to Theorem 2.3 and an argument on the treewidth of the multicut duals for this instance. Formally, we prove the following result, where $\tau(\mathcal{P})$ denotes the sum of the number of components of $G_0[Y]$ over all $Y \in \mathcal{P}$.

► **Lemma 3.4.** *Let (G, H) be a connected planar + πe instance of MULTICUT and let a complete maximum valid state \mathcal{P} of (G, H) with $\mathcal{P}_{\text{thin}} = \emptyset$ be given. Then a minimum multicut for (G, H) can be computed in $f(\pi, \tau(\mathcal{P}))n^{O(\sqrt{\mu})}$.*

It remains to show that we can design our procedure so that we reach a complete state sufficiently fast. In order to keep track of the progress of our algorithm, we need a measure of how far advanced a given state is. To this end, given a state \mathcal{P} , we first define an integer $\kappa(\mathcal{P}, Y)$ for every $Y \in \mathcal{P}$ by

$$\kappa(\mathcal{P}, Y) = \begin{cases} 2(\pi(t+1) + 1), & \text{for } Y \in \mathcal{P}_{\text{fat}}, \\ \pi(t+1) + 1 + \alpha_{G_0, \mathcal{P}}(Y), & \text{for } Y \in \mathcal{P}_{\text{f-n}}, \\ \alpha_{G_0, \mathcal{P}}(Y), & \text{for } Y \in \mathcal{P}_{\text{thin}}. \end{cases}$$

Now, we define $\kappa(\mathcal{P}) = \sum_{Y \in \mathcal{P}} \kappa(\mathcal{P}, Y)$.

Our objective is to make the value of κ increase in each iteration of the algorithm. In order to make this possible, we need that the addition of a vertex to a given set in the state makes the value of α of this set increase. In order for that to hold, we need the classes in the state to be as large as possible in a certain sense. More precisely, given a graph G and two disjoint sets $Y_1, Y_2 \subseteq V(G)$, we say that a set Y_3 is *relevant* for (Y_1, Y_2) in G if $Y_1 \subseteq Y_3, Y_2 \cap Y_3 = \emptyset, d_G(Y_3) = \lambda_G(Y_1, Y_2)$ and Y_3 is maximum among all sets with these properties. Further, we say that a state \mathcal{P} is *relevant* if for all $Y \in \mathcal{P}$, we have that Y is a relevant set for $(Y, \bigcup \mathcal{P} \setminus Y)$. Luckily, the following result whose proof needs some careful reconfiguration arguments, allows us to turn arbitrary states in consideration into relevant ones without dropping any important properties.

► **Lemma 3.5.** *Let a connected planar + πe instance (G, H) of MULTICUT and a state \mathcal{P} of (G, H) be given. Then, in polynomial time, we can compute a relevant state \mathcal{P}' of (G, H) such that $|\mathcal{P}'| = |\mathcal{P}|, \tau(\mathcal{P}') \leq \tau(\mathcal{P}), \kappa(\mathcal{P}') \geq \kappa(\mathcal{P})$ and, if \mathcal{P} is maximum valid, then so is \mathcal{P}' .*

With this result at hand, we can handle incomplete states. As pointed out before, given an incomplete relevant state \mathcal{P} , we choose a vertex $u_0 \in V(G) \setminus \bigcup \mathcal{P}$ that has a neighbor in a thin class Y of \mathcal{P} . We now consider all states that are obtained by adding u_0 to one of these partition classes. If u_0 is added to a thin or a fat-neighboring partition class, then, as \mathcal{P} is relevant, the value of α increases for this class and hence κ increases. If u_0 is added to a fat class, then Y becomes fat-neighboring and hence κ also increases. This shows that a complete state can be reached sufficiently fast. In order to apply Lemma 3.4, we also need to make sure that $\tau(\mathcal{P})$ does not increase too fast. However, this follows readily by construction. Our handling of relevant incomplete states is subsumed in the following result.

► **Lemma 3.6.** *Let (G, H) be a connected planar + πe instance of MULTICUT and let a relevant incomplete state \mathcal{P} of (G, H) be given. Then, in polynomial time, we can compute a collection $(\mathcal{P}^i)_{i \in [q]}$ of q states of (G, H) for some $q \leq \pi + t$ such that $\tau(\mathcal{P}^i) \leq \tau(\mathcal{P}) + 1$ and $\kappa(\mathcal{P}^i) > \kappa(\mathcal{P})$ hold for all $i \in [q]$ and if \mathcal{P} is maximum valid, then there exists some $i \in [q]$ such that \mathcal{P}^i is maximum valid.*

The overall strategy for handling Lemma 3.1 is now to apply Lemma 3.5 and Lemma 3.6 until we are left with a collection of complete states. These can then be handled by Lemma 3.3 and Lemma 3.4.

4 Crossing number

In this section, a *drawing* of a graph G consists of a mapping of the vertices of G to points in the plane and a mapping of the edges of G to curves in the plane connecting their endpoints such that all vertices and edges of G are in general position except the required adjacencies and such that in any point not corresponding to a vertex, at most two edges intersect. An *edge crossing* refers to a point that does not correspond to a vertex and in which two edges intersect. Given an instance (G, H) of MULTICUT we use cr for the *crossing number* of a graph G , which is defined to be the smallest integer k such that there exists a drawing of G with exactly k edge crossings. The following is the main result of this section.

► **Theorem 1.7.** *Edge-weighted MULTICUT can be solved in time $f(cr, t)n^{O(\sqrt{\mu})}$.*

In order to make our algorithm work, rather than just the crossing number of a graph, we also need to have an optimal drawing available. This can be achieved through the following result due to Grohe [7].

► **Proposition 4.1.** *Given a graph G , there exists an algorithm that computes a drawing of G in the plane minimizing the number of crossings and that runs in time $f(cr)n^{O(1)}$.*

Formally, an instance (G, H) of MULTICUT is called *crossing-planar* if G is given as a drawing of a graph. Given a crossing-planar instance, we denote by E_{cr} the set of edges in $E(G)$ crossing at least one other edge of $E(G)$ and we set $\overline{cr} = |E_{cr}|$. Observe that \overline{cr} can be arbitrarily large in comparison to cr as \overline{cr} refers to the given drawing and cr to an optimal drawing of the abstract graph G . However, when considering an optimal drawing of G , we have $\overline{cr} \leq 2cr$ as every crossing involves 2 edges.

For our algorithm, it will be convenient to assume that the instances in consideration satisfy a technical extra condition. Namely, a crossing-planar instance (G, H) of MULTICUT is called *normalized* if every $e \in E_{cr}$ is of infinite weight. Observe that for any normalized crossing-planar instance (G, H) of MULTICUT, any finite multicut S of (G, H) and any $e \in E_{cr}$, we have that $V(e)$ is contained in one component of $G \setminus S$. Our main technical contribution is proving the restriction of Theorem 1.7 to normalized instances.

► **Theorem 4.2.** *There exists an algorithm that solves every normalized crossing-planar instance of MULTICUT and runs in $f(\overline{cr}, t)n^{O(\sqrt{\mu})}$.*

The simple proof that Theorem 4.2 implies Theorem 1.7 is postponed to the full version. It remains to prove Theorem 4.2.

Given a normalized crossing-planar instance (G, H) of MULTICUT, we denote by \mathcal{H} the set of graphs H' on $V(H) \cup V(E_{cr})$ and we use G' for $G \setminus E_{cr}$. Further, we let F^* denote the faces of G' which contain a pair of crossing edges in G . The main technical difficulty for the proof of Theorem 4.2 is contained in the following result.

► **Lemma 4.3.** *Let (G, H) be a normalized crossing-planar instance of MULTICUT admitting a multicut of finite weight. Then there exists $H' \in \mathcal{H}$ such that every minimum weight multicut for (G', H') is a minimum weight multicut for (G, H) and for every subcubic, inclusion-wise minimal minimum weight multicut dual C for (G', H') , we have $\text{tw}(C - F^*) = O(\sqrt{\mu})$.*

Due to space restrictions, the full proof of Lemma 4.3 is postponed to the full version. We now give an overview of the proof of Lemma 4.3.

Let (B_1, B_2, I, X) be an extended biclique decomposition of H with $|X| = \mu$. Further, let S^* be a minimum weight multicut for (G, H) such that S^* minimizes the number of elements of $V(H) \cup V(E_{cr})$ that are in a component of $G \setminus S^*$ that also contains a terminal of X . In the following, we use \overline{X} for the set of elements of $V(H) \cup V(E_{cr})$ that are in a component of $G \setminus S^*$ that also contains a terminal of X , we use $\overline{B_1}$ for the set of elements of $(V(H) \cup V(E_{cr})) \setminus \overline{X}$ that are in a component of $G \setminus S^*$ that also contains a terminal of B_1 , and we use $\overline{B_2}$ for $(V(H) \cup V(E_{cr})) \setminus (\overline{X} \cup \overline{B_1})$. Observe that $(\overline{B_1}, \overline{B_2}, \overline{X})$ is a partition of $V(H) \cup V(E_{cr})$. We now define H' to be the complete multipartite graph on $V(H) \cup V(E_{cr})$ such that $\overline{B_1}$ and $\overline{B_2}$ are partition classes and such that for every component of $G \setminus S^*$ that contains at least one terminal of X , all the elements of \overline{X} contained in this component form a partition class.

This finishes the description of H' . Observe that $H' \in \mathcal{H}$. Further observe that H is a subgraph of H' . The following two claims directly imply that every minimum weight multicut for (G', H') is also a minimum weight multicut for (G, H) . Claim 4.4 will be reused for the proof of the treewidth bound. These claims mainly follow from the construction of (G', H') .

▷ **Claim 4.4.** Let S' be a multicut for (G', H') . Then S' is also a multicut for (G, H) .

▷ **Claim 4.5.** S^* is a multicut for (G', H') .

We are now ready to conclude that any minimum weight multicut S for (G', H') is a minimum weight multicut for (G, H) . First observe that S is a multicut for (G, H) by Claim 4.4 and as H is a subgraph of H' . Further, by Claim 4.5 and the minimality of S , we obtain $w(S) \leq w(S^*)$ where w is the weight function associated to G . By the minimality of S^* , the statement follows.

In the following, let C be a subcubic, inclusion-wise minimal, minimum weight multicut dual for (G', H') . We will give the desired treewidth bound on C . Let $S' = E_G(C)$ and observe that S' is a minimum weight multicut for (G', H') by Lemma 2.1. It hence follows from the first part of the lemma that S' is also a minimum weight multicut for (G, H) . Further, we use C' for $C - F^*$.

Our strategy for proving the second part of the lemma is to show that the faces of C' containing a terminal of \bar{X} form a small set of faces that is close to every face of C' in a certain sense. We first give the following result showing that the number of such faces is small indeed. Its proof uses the fact that S^* is chosen so that the size of \bar{X} is minimized and the fact that S' is a minimum weight multicut for (G, H) .

▷ **Claim 4.6.** Let f be a face of C' that contains a terminal of \bar{X} . Then f also contains a terminal of X .

The next result uses the structure of H .

▷ **Claim 4.7.** Let $v \in V(C')$ with $d_{C'}(v) = 3$. Then v is incident to a face of C' that contains a terminal of \bar{X} .

We are now ready to conclude the desired treewidth bound on C' . Let C'' be a component of C' . For every face f of C'' , we now introduce a vertex z_f that shares an edge with every vertex that is incident to f . Let C''_0 be the resulting graph. It turns out that C''_0 is a planar graph and the vertices corresponding to faces containing terminals of \bar{X} form a 3-dominating set in C''_0 . By Proposition 2.2 and $|A| \leq |X| \leq \mu$, we obtain that $\text{tw}(C''_0) = O(\sqrt{\mu})$ and hence $\text{tw}(C') = O(\sqrt{\mu})$. This allows to conclude Lemma 4.3.

With Lemma 4.3 at hand, it is easy to conclude Theorem 4.2. The proof can be found in the full version.

References

- 1 Vincent Cohen-Addad, Éric Colin de Verdière, Dániel Marx, and Arnaud de Mesmay. Almost tight lower bounds for hard cutting problems in embedded graphs. *J. ACM*, 68(4):30:1–30:26, 2021. doi:10.1145/3450704.
- 2 Éric Colin de Verdière. Multicuts in planar and bounded-genus graphs with bounded number of terminals. *Algorithmica*, 78:1206–1224, 2017. doi:10.1007/S00453-016-0258-0.
- 3 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. doi:10.1137/S0097539792225297.

- 4 Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Fixed-parameter algorithms for (k, r) -center in planar graphs and map graphs. *ACM Trans. Algorithms*, 1(1):33–47, July 2005. doi:10.1145/1077464.1077468.
- 5 Jacob Focke, Florian Hörsch, Shaohua Li, and Dániel Marx. Multicut Problems in Embedded Graphs: The Dependency of Complexity on the Demand Pattern. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry (SoCG 2024)*, volume 293 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SocG.2024.57.
- 6 Jacob Focke, Florian Hörsch, Shaohua Li, and Dániel Marx. Multicut problems in embedded graphs: The dependency of complexity on the demand pattern, 2025. arXiv:2312.11086.
- 7 Martin Grohe. Computing crossing numbers in quadratic time. *Journal of Computer and System Sciences*, 68(2):285–302, 2004. Special Issue on STOC 2001. doi:10.1016/j.jcss.2003.07.008.
- 8 T. C. Hu. Multi-commodity network flows. *Operations Research*, 11(3):344–360, 1963.
- 9 Florian Hörsch and Dániel Marx. Multicut problems in almost-planar graphs: The dependency of complexity on the demand pattern, 2025. doi:10.48550/arXiv.2504.21624.
- 10 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, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 569–580. Springer, 2012. doi:10.1007/978-3-642-31594-7_48.
- 11 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, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 677–688. Springer, 2012. doi:10.1007/978-3-642-31594-7_57.
- 12 Paul D. Seymour. A short proof of the two-commodity flow theorem. *J. Comb. Theory, Ser. B*, 26(3):370–371, 1979. doi:10.1016/0095-8956(79)90012-1.