Multicut Problems in Embedded Graphs: The Dependency of Complexity on the Demand Pattern

Jacob Focke
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Florian Hörsch
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Shaohua Li
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Dániel Marx
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Abstract

The Multicut problem asks for a minimum cut separating certain pairs of vertices: formally, given a graph G and a demand graph H on a set $T \subseteq V(G)$ of terminals, the task is to find a minimum-weight set C of edges of G such that whenever two vertices of T are adjacent in H, they are in different components of $G \setminus C$. Colin de Verdière [*Algorithmica,* 2017] showed that Multicut with t terminals on a graph G of genus g can be solved in time $f(t,g)n^{O(\sqrt{g^2+gt+t})}$. Cohen-Addad et al. [*JACM*, 2021] proved a matching lower bound showing that the exponent of n is essentially best possible (for every fixed value of t and g), even in the special case of Multiway Cut, where the demand graph H is a complete graph.

However, this lower bound tells us nothing about other special cases of Multicut such as Group 3-Terminal Cut (where three groups of terminals need to be separated from each other). We show that if the demand pattern is, in some sense, close to being a complete bipartite graph, then Multicut can be solved faster than $f(t,g)n^{O(\sqrt{g^2+gt+t})}$, and furthermore this is the only property that allows such an improvement. Formally, for a class $\mathcal{H}$ of graphs, Multicut($\mathcal{H}$) is the special case where the demand graph H is in $\mathcal{H}$. For every fixed class $\mathcal{H}$ (satisfying some mild closure property), fixed g, and fixed t, our main result gives tight upper and lower bounds on the exponent of n in algorithms solving Multicut($\mathcal{H}$).

In addition, we investigate a similar setting where, instead of parameterizing by the genus g of G, we parameterize by the minimum number k of edges of G that need to be deleted to obtain a planar graph. Interestingly, in this setting it makes a significant difference whether the graph G is weighted or unweighted: further nontrivial algorithmic techniques give substantial improvements in the unweighted case.

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, Genus, Surface, Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.57

Related Version *Full Version*: `https://arxiv.org/abs/2312.11086` [9]

1 Introduction

Computing cuts and flows in graphs is a fundamental problem in theoretical computer science, with algorithms going back to the early years of combinatorial optimization [8, 12] and significant new developments appearing even in recent years [35, 36]. Given two vertices s and t in a weighted graph G, an $s - t$ *cut* is a set S of edges such that $G \setminus S$ has no path

40th International Symposium on Computational Geometry (SoCG 2024).
Editors: Wolfgang Mulzer and Jeff M. Phillips; Article No. 57; pp. 57:1–57:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

connecting s and t. It is well known that an $s-t$ cut of minimum total weight can be found in polynomial time. However, the problem becomes much harder when generalized to more than two terminals. In the Multiway Cut problem, we are given a graph G and a set $T \subseteq V(G)$ of *terminals*, and the task is to find a *multiway cut* S of minimum weight, that is, a set $S \subseteq E(G)$ such that every component of $G \setminus S$ contains at most one terminal. Dahlhaus et al. [7] showed that Multiway Cut is NP-hard even for three terminals.

The study of algorithms on planar graphs is motivated by the fact that planar graphs can be considered a theoretical model of road networks or other physical networks where crossing of edges can be assumed to be unlikely. While most NP-hard problems remain NP-hard on planar graphs, there is often some computational advantage that can be gained by exploiting planarity. In many cases, this advantage takes the form of a square root appearing in the running time of an algorithm soving the problem [3,5,6,11,21–23,28–30,32]. Multiway Cut remains NP-hard on planar graphs, but unlike in general graphs, where the problem is hard for three terminals, the problem can be solved in polynomial time on planar graphs for any fixed number of terminals. Dahlhaus et al. [7] gave an algorithm with a running time of the form $f(t)n^{O(t)}$ for t terminals, which was improved to $n^{O(\sqrt{t})}$ by Colin de Verdière [5] and Klein and Marx [21]. The square root in the exponent appears to be best possible: Marx [28] showed that, assuming the Exponential-Time Hypothesis (ETH), there is no algorithm for Multiway Cut on planar graphs with running time $f(t)n^{o(\sqrt{t})}$ for any computable f.

Colin de Verdière [5] actually showed a much stronger result than just an $f(t)n^{O(\sqrt{t})}$ algorithm for Multiway Cut on planar graphs with t terminals. First, the algorithm works not only on planar graphs, but on graphs that can be embedded on a surface of genus at most g for some fixed constant g. The second notable feature of the algorithm of Colin de Verdière [5] is that it considers the generalization Multicut of Multiway Cut, where instead of requiring that all terminals have to be disconnected from each other, the input contains a demand pattern describing which pairs of terminals have to be disconnected.

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.

Observe that the special case of Multicut when H is a complete graph on T is exactly the same as Multiway Cut. Colin de Verdière [5] showed that the running time achieved for Multiway Cut can be obtained also for Multicut in its full generality with arbitrary demand pattern.

▶ **Theorem 1** (Colin de Verdière [5]). Multicut *can be solved in time $f(g,t)n^{O(\sqrt{g^2+gt+t})}$ for some function f, where g is the Euler genus of G and t is the number of terminals.*

In the following, we adopt the convention that if an instance (G,H) of Multicut is clear from the context, then we use $t = |T|$ for the number of terminals, g for the Euler genus of G, $\vec{g}$ for the orientable genus of G, and $n = |V(G)|$ for the number of vertices of G. Furthermore, if not otherwise specified, a function f appearing in the running time means that there exists a computable function f that can serve as the runtime bound.

The exponent $O(\sqrt{g^2+gt+t})$ in Theorem 1 might look somewhat arbitrary. However, Cohen-Addad et al. [4] showed that this form of the exponent is optimal under ETH.

▶ **Theorem 2** (Cohen-Addad et al. [4]). *Assuming the* ETH*, there exists a universal constant* $\alpha \geq 0$ *such that for any fixed choice of integers* $\vec{g} \geq 0$ *and* $t \geq 4$*, there is no algorithm that decides all unweighted* Multiway Cut *instances with orientable genus at most* $\vec{g}$ *and at most* t *terminals, in time* $O(n^{\alpha\sqrt{\vec{g}^2+\vec{g}t+t}/\log(\vec{g}+t)})$.

Note that saying orientable genus in Theorem 2 makes the lower bound statement stronger: if we consider Euler genus instead, then we would consider a parameter that is always at most two times larger than orientable genus, hence we would obtain the same lower bound (with a factor-2 loss in the exponent, which can be accounted for by reducing α by a factor of 2). In similar way, the statement implies an analogous lower bound for nonorientable surfaces. The lower bounds below (for Multiway Cut and Multicut) will be formulated in terms of orientable genus, as this is the strongest form of the lower bound.

1.1 Our Contributions

Bounded-genus graphs. While Theorem 2 appears to give a fairly tight lower bound, showing the optimality of Theorem 1 in every parameter range, it has its shortcomings. First, as a minor issue, it works only for $t \geq 4$ (this will be resolved in the current paper). More importantly, while Theorem 1 solves the Multicut problem in its full generality, Theorem 2 gives a lower bound for the special case Multiway Cut (that is, when H is a clique). This is a general issue when interpreting tight bounds: algorithmic results solve all instances of a problem, while lower bounds show the hardness of a specific type of instances. This leaves open the possibility that there exist nontrivial classes of instances ("islands of tractability") that can be solved significantly faster than the known upper bound. In our case, Theorem 2 shows that Theorem 1 is optimal in the case when the demand graph H is a clique. The main goal of this paper is to understand whether Theorem 1 is optimal for other classes of patterns as well.

Are there nontrivial classes of demand pattern graphs where the algorithm of Theorem 1 is not optimal, that is, the exponent can be better than $O(\sqrt{g^2+gt+t})$?

There is a simple, trivial case when Theorem 1 is not optimal: if the demand graph is a biclique (complete bipartite graph), then the Multicut instance can be solved in polynomial time (as then the task reduces to separating two sets T_1 and T_2 of terminals from each other). But what happens if we consider, say, demand patterns that are complete tripartite graphs? This special case is also known as Group 3-Terminal Cut: 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 $i \neq j$. Does the lower bound of Theorem 2 for Multiway Cut hold for Group 3-Terminal Cut as well? Previous work does not give an answer to this question. One can imagine many other classes of demand patterns where we have no answer yet, for example, patterns obtained as the disjoint union of a biclique and a triangle. Our goal is to exhaustively investigate *every* such pattern class and provide an answer for each of them.

Formally, let $\mathcal{H}$ be a class of graphs. Then we denote by 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 Multicut($\mathcal{H}$) is exactly Multiway Cut. The formal question that we are investigating is if, for a given $\mathcal{H}$, Theorem 1 gives an optimal algorithm for Multicut($\mathcal{H}$).

Let us first ask a simpler qualitative question: in an algorithm for Multicut($\mathcal{H}$), do parameters g and/or t have to appear in the exponent of n? The notion of W[1]-hardness is an appropriate tool to answer this question. A problem is said to be *fixed-parameter tractable (FPT)* with respect to some parameter k appearing in the input if there is an algorithm with

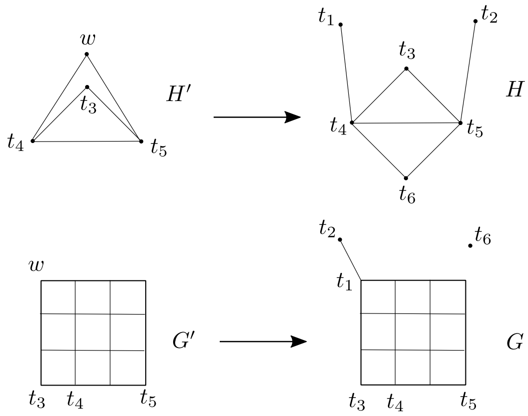

Figure 1 Reducing a Multicut instance (G', H') to (G, H) where H' is a projection of H.

running time $f(k)n^{O(1)}$ for some computable function f [6]. If a problem is W[1]-hard, then this is interpreted as strong evidence that it is not FPT, which means that the parameter has to appear in the exponent of n.

If H is a biclique, then Multicut corresponds to the problem of separating two sets of terminals, which can be reduced to a minimum $s - t$ cut problem. If H has only two edges, then a similar reduction is known [16, 33]. 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. We show that in case of nontrivial patterns, the genus has to appear in the exponent.

▶ **Theorem 4.** *Let $\mathcal{H}$ be a computable class of graphs.*
1. *If every graph in $\mathcal{H}$ is a trivial pattern, then* Multicut($\mathcal{H}$) *is polynomial-time solvable.*
2. *Otherwise,* Multicut($\mathcal{H}$) *is* W[1]*-hard parameterized by $\vec{g}$, even when restricted to instances for which the number of terminals is bounded by some constant depending on $\mathcal{H}$.*

For the parameter t, the number of terminals, we find nontrivial cases where it does *not* have to appear in the exponent of the running time. 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.

▶ **Theorem 5.** *Let $\mathcal{H}$ be a computable class of graphs.*
1. *If $\mathcal{H}$ has bounded distance to extended bicliques, then* Multicut($\mathcal{H}$) *can be solved in time $f(t, g)n^{O(g)}$.*
2. *Otherwise,* Multicut($\mathcal{H}$) *is* W[1]*-hard parameterized by t, even on planar graphs.*

The lower bounds in Theorem 4 and Theorem 5 start with the following simple observation. Let H be a graph and let H' be obtained by identifying two nonadjacent vertices u and v in H into a new vertex w. We observe that an instance (G', H') of Multicut can be reduced to some instance (G, H) in a very simple way: let us obtain G by replacing the vertex w in G' with u, and connect to u a new vertex v with an edge of sufficiently large weight.

Intuitively, this means that the demand pattern H' is not harder than the pattern H, so if $\mathcal{H}$ contains H, then we might as well assume that $\mathcal{H}$ contains H'. More formally, in the case of W[1]-hardness, it can be shown that adding H' to $\mathcal{H}$ does not change whether Multicut($\mathcal{H}$)

is W[1]-hard. We can argue similarly about the graph H' obtained by removing a vertex v of H (in this case, the reduction simply extends G' with the isolated vertex v). 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 (see Figure 1). 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. Due to the observations above, it is sufficient to show the W[1]-hardness results of Theorem 4 and Theorem 5 for projection-closed classes $\mathcal{H}$. In particular, the proof of Theorem 5 uses Ramsey-theoretic arguments to show that a projection-closed class either has bounded distance to extended bicliques, or contains all cliques, or contains all complete tripartite graphs. Thus, the hardness results essentially boil down to two cases: cliques (i.e., Multiway Cut) and complete tripartite graphs (i.e., Group 3-Terminal Cut).

Let us turn our attention now to quantitative lower bounds of the form of Theorem 2. Unfortunately, the reduction from (G', H') to (G, H) increases the number t of terminals, hence formally cannot be used to obtain lower bounds on the exact dependence on t. Nevertheless, one feels that this reduction should be somehow "free." To obtain a more robust setting, we express this feeling by restricting our attention to projection-closed classes. For such classes $\mathcal{H}$, we obtain tight bounds showing that Multicut($\mathcal{H}$) can have two types of behavior, depending on whether $\mathcal{H}$ has bounded distance to extended bicliques.

▶ **Theorem 6.** *Let $\mathcal{H}$ be a computable projection-closed class of graphs. Then the following holds for* 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.*

Statement 3(a) follows from Theorem 1, while statement 2(a) is a new nontrivial algorithmic statement. Observe that in the special case when $\mathcal{H}$ is all cliques, statement 3(b) recovers Theorem 2 on Multiway Cut, with the strengthening that it works also for $t = 3$, which was previously left as an open problem by Cohen-Addad et al. [4].

▶ **Corollary 7.** *Assuming* ETH*, there exists a universal constant $\alpha > 0$ such that for any fixed choice of integers $\vec{g} \geq 0$ and $t \geq 3$, there is no algorithm that decides all unweighted* Multiway Cut *instances with orientable genus at most $\vec{g}$ and at most t terminals, in time $O(n^{\alpha\sqrt{\vec{g}^2+\vec{g}t+t}/\log(\vec{g}+t)})$.*

Let us remark that, in Theorem 6, one reason to require the condition that $\mathcal{H}$ be projection-closed is to avoid instances where most terminals are artificial and do not play any role. For example, consider the class $\mathcal{H}$ that contains all graphs H that consist of a clique of size $\log(|V(H)|)$ and a collection of isolated vertices. This means that Multicut($\mathcal{H}$) is equivalent to solving Multiway Cut on the clique part and ignoring the isolated vertices. Intuitively, this suggests that Multicut($\mathcal{H}$) should not be easier than Multiway Cut, but it will turn out, see Theorem 12, that there is an algorithm solving Multicut($\mathcal{H}$) whose running time is better than the lower bound in Theorem 6 3(b).

Planar graphs with extra edges. One can argue that even if a graph has bounded genus, this can mean a fairly significant violation of planarity: arbitrarily many edges can go through a handle, creating many connections between two distant parts of the graph. Let us study a graph parameter that models a much smaller deviation from planarity. Given a graph class $\mathcal{F}$, Cai [2] introduced the class $\mathcal{F} + \pi e$ that contains every graph that can be made a member of $\mathcal{F}$ by the removal of at most π edges. If some algorithmic problem is easy or well understood on a graph class $\mathcal{F}$, then we can explore how it gets harder as we move away from this class by considering $\mathcal{F} + \pi e$ for larger and larger π. This viewpoint of parameterizing by the "distance from triviality" has been studied in the combination of different algorithmic problems, graph classes, and distance measures [1, 10, 13–15, 17, 18, 24, 25, 34].

In the following, we denote by $\mathsf{planar} + \pi e$ the class of graphs that can be made planar by the removal of at most π edges. As a convention, given an instance (G, H), we will denote by π the smallest integer such that the graph is in $\mathsf{planar} + \pi e$.

So far, we have mostly ignored the question of whether the graph G is weighted or unweighted. If G has polynomially large integer weights, then it is easy to reduce the problem to unweighted graphs with a transformation that does not change genus. However, it is not clear at all if an edge-weighted problem on a graph in $\mathsf{planar} + \pi e$ can be reduced to an unweighted problem without significantly increasing the distance from planarity.

For weighted $\mathsf{planar} + \pi e$ graphs, we get similar results as for bounded-genus graphs, but with quantitative bounds of somewhat different form.

▶ **Theorem 8.** *Let $\mathcal{H}$ be a computable class of graphs.*

1. *If every graph in $\mathcal{H}$ is a trivial pattern, then edge-weighted* Multicut($\mathcal{H}$) *is polynomial-time solvable.*
2. *Otherwise, edge-weighted* Multicut($\mathcal{H}$) *is* W[1]*-hard parameterized by π, even when restricted to instances for which the number of terminals is bounded by some constant depending on $\mathcal{H}$.*

We also prove the analog of Theorem 2, where the optimal exponent is now $O(\sqrt{\pi})$ instead of $O(g)$, and $O(\sqrt{\pi + t})$ instead of $O(\sqrt{g^2 + gt + t})$.

▶ **Theorem 9.** *Let $\mathcal{H}$ be a computable projection-closed class of graphs. Then the following holds for edge-weighted* 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* $\mathsf{planar} + \pi e$ *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* $\mathsf{planar} + \pi e$ *instances with at most t terminals.*

In unweighted graphs, the situation is different and we can achieve better running times.

▶ **Theorem 10.** *Unweighted* Multicut *can be solved in time $f(\pi, t)n^{O(\sqrt{t})}$.*

1.2 Our Techniques

Our contributions consist of upper bounds, combinatorial results, and lower bounds.

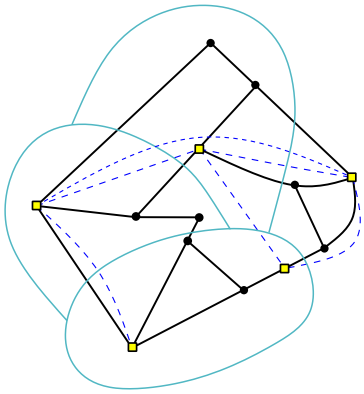

Figure 2 An example of a multicut dual. The yellow squares represent the terminals. The blue dashed curves represent the edges of H. The cyan graph represents a multicut dual.

Algorithmic results. Similarly to earlier work [5], we may assume that if the input graph G has Euler genus g, then we have an embedding of G into a surface N of Euler genus g: there are algorithms for finding such an embedding whose running time is dominated by the time bound we want to achieve [20, 31]. We say that an instance (G, H) of Multicut is *N-embedded* if G is given in the form of a graph embedded in a surface N of Euler genus g. The main insight behind the algorithmic results for Multicut is a simple consequence of duality: if $S \subseteq E(G)$ is a solution in a connected graph G, then in the dual graph G^* the edges of S create a graph where the terminals that need to be separated are in different faces (that is, any curve connecting them crosses some edge of S in G^*). This motivates the following definition: a *multicut dual* for (G, H) is a graph C embedded on N that is in general position with G and, for every $t_1, t_2 \in V(H)$ with $t_1t_2 \in E(H)$, it holds that t_1 and t_2 are contained in different faces of C (see Figure 2). Here "general position" means that the drawings of C and G intersect in finitely many points, none of which is a vertex of either graph. We define the *weight* $w(C)$ of a multicut dual to be the total weight of the edges of G crossed by C. The minimum weight of a multicut for (G, H) is the same as the minimum weight of a multicut dual.

The algorithm of Theorem 1 proceeds by guessing the topology of an optimal multicut dual, finding a minimum-weight multicut dual C with this topology, and then arguing that the edges crossed by C form indeed a minimum solution of the Multicut instance (G, H). It is argued that the multicut dual C has $O(t + g)$ vertices, which implies that it has pathwidth $O(\sqrt{g^2 + gt + t})$. The exponent of n in the algorithm is determined by this pathwidth bound.

One can observe that if we have some other way of bounding the pathwidth of the multicut dual C (perhaps because of some extra conditions on the instance (G, H)), then the algorithm can be made to run in time with that pathwidth bound in the exponent. Furthermore, by modifying the algorithm, we can also use a bound on treewidth instead of pathwidth. The following statement can be extracted from the proof of Theorem 1:

▶ **Theorem 11.** *On N-embedded instances (G, H) of* Multicut *for which every minimum-weight inclusionwise minimal multicut dual has treewidth at most β,* Multicut *can be solved in time $f(t, g)n^{O(\beta)}$.*

We believe that formulating Theorem 11 gives a useful tool: as we will see, it allows more efficient algorithms in certain cases. Let us quickly remark that Theorem 11 can be used to recover Theorem 1. It can be shown that every face of an inclusionwise minimal multicut

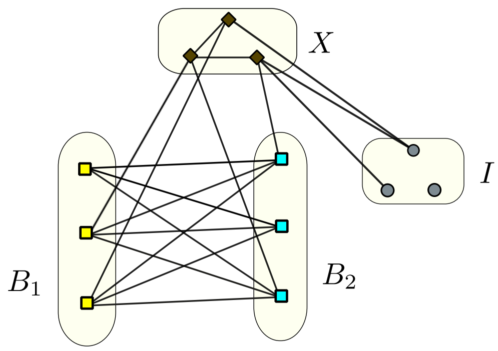

Figure 3 An example of an extended biclique partition.

contains a terminal and that a graph with at most t faces drawn on a surface of Euler genus g has treewidth $O(\sqrt{g^2+gt+t})$. Thus if there are t terminals, then every minimum-weight inclusionwise minimal multicut dual has treewidth $O(\sqrt{g^2+gt+t})$, and then Theorem 11 gives the required running time.

Our first application of Theorem 11 is to prove Theorem 6 (2a), the algorithm for MULTICUT($\mathcal{H}$) when $\mathcal{H}$ has bounded distance to extended bicliques.

▶ **Theorem 12.** *Let $\mathcal{H}$ be a class of graphs whose distance to extended bicliques is at most μ. Then* MULTICUT($\mathcal{H}$) *can be solved in time $f(t,g)n^{O(\mu+g\mu)}$.*

An *extended biclique partition* of a graph H is a partition (B_1, B_2, I, X) of its vertices such that $H - X$ is an extended biclique with a complete bipartite graph with bipartition (B_1, B_2) and isolated vertices X (see Figure 3). Consider an instance (G, H) and an extended biclique partition (B_1, B_2, I, X) of H with $|X|$ being at most some constant μ. Let us consider a minimum-weight inclusionwise minimal multicut dual C (see Figure 4). There are at most μ faces of C that contain a terminal of X. Let us consider a face F that is not adjacent to any of these at most μ faces. Let Z be the set of terminals in face F. If $Z \subseteq I$, then we can remove all the edges of C between F and a neighboring face F': this would create only connections between I and $B_1 \cup B_2 \cup I$, which are not forbidden. Thus this would contradict the inclusionwise minimality of C. Assume, without loss of generality, that $Z \cap B_1 \neq \emptyset$. With a similar argument, we can assume that if Z' is the set of terminals in the faces adjacent to F, then $Z' \subseteq B_2 \cup I$.

The key observation is the following. Let us consider the set M of edges of C on the boundary of F. What can go wrong if we omit M from the multicut dual, i.e., consider $C \setminus M$? Then F gets merged with the adjacent faces, which creates two kinds of new connections between the terminals: (1) between terminals in Z and Z', and (2) between terminals in Z'. However, as $Z' \subseteq B_2 \cup I$, which is an independent set in H, new connections inside Z' are not a problem. Furthermore, connections between Z and Z' can be resolved by extending the solutions given by $C \setminus M$ with a minimum cut separating Z from the rest of the terminals. As in particular M gives such a cut, the weight of this modified solution is not larger than the original one. That is, the optimal multicut for (G, H) can be decomposed into a solution S_1 for the problem $(G, H - Z)$ and a minimum cut S_2 separating Z and $V(H) \setminus Z$.

This suggests the following win/win approach. Let D be the set of at most μ faces of the optimal multicut dual C that contains terminals from X. If there is a face of C that is not adjacent to a face in D, then a decomposition is possible. Otherwise, D is a dominating set of size at most μ in the dual of C, which implies, using known results on bounded-genus

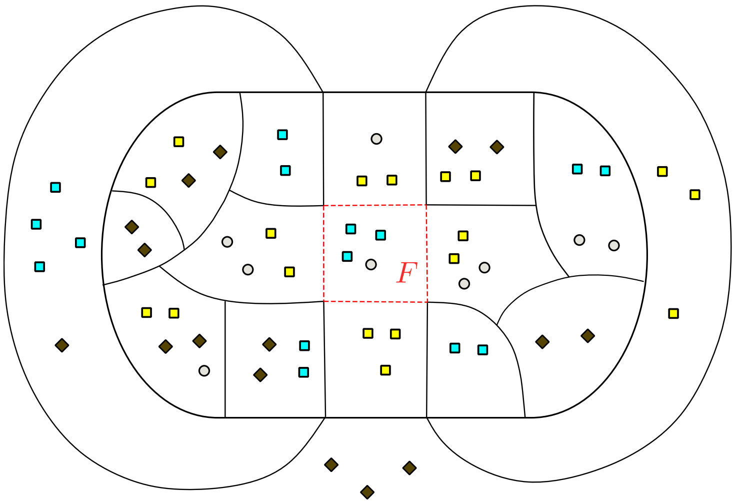

Figure 4 An example of a Z-reducible instance. The terminals in B_1, B_2, I, and X are represented by yellow, blue, gray, and rotated dark squares, respectively. The rotated dark squares represent the terminals of X. Let F be the face whose adjacent faces do not contain a terminal from X and let Z be the terminals of F.

graphs, a bound on the treewidth of C. Formally, we say that (G, H) is *Z-reducible* if for every minimum-weight $(Z, V(H) - Z)$-cut S_1 and every minimum-weight multicut S_2 of $(G, H - Z)$, we have that $S_1 \cup S_2$ is a minimum-weight multicut of (G, H).

▶ **Lemma 13.** *Let (G, H) be an N-embedded instance of* Multicut *and let (B_1, B_2, I, X) be an extended biclique partition of $V(H)$ with $|X| \leq \mu$. Then one of the following holds:*
(1) *there is some $i \in [2]$ and $Z \subseteq B_i \cup I$ such that (G, H) is Z-reducible,*
(2) $\operatorname{tw}(C) = O(\mu + g\mu)$ *for every minimum-weight inclusionwise minimal multicut dual C for (G, H).*

As a second application of Theorem 11, let us sketch the proof of Theorem 10: an algorithm for unweighted planar $+ \pi e$ graphs where the exponent of the running time does not contain π. Let (G, H) be an instance of Multicut with t terminals and let E_π be a set of π edges such that $G \setminus E_\pi$ is planar. Let $G_0 = G \setminus E_\pi$ and let W be the endpoints of E_π. First, we guess how, for some hypothetical optimum solution S, the components of $G_0 \setminus S$ partition $T \cup W$ (there are $O(2^{(t+\pi)\log(t+\pi)})$ such partitions). Let $(P_1, \ldots, P_\ell)$ be this partition and let H' be the corresponding ℓ-partite graph on $T \cup W$. Let $E_0 \subseteq E_\pi$ contain those edges for which both endpoints are in the same P_i. Clearly, every solution for (G, H') should include every edge of $E_\pi \setminus E_0$ (as they go between two different P_i's) and the edges in E_0 do not change the solution. Thus we can ignore the edges in E_π and solve the instance (G_0, H') on the planar graph G_0. If we try to solve this instance using Theorem 1, then the exponent of the running time would be $O(\sqrt{t + \pi})$ as we now have $|T \cup W| = O(t + \pi)$ terminals. Our goal is to preprocess this instance such that if S' is an optimum solution of (G_0, H'), then every component of $G_0 \setminus S'$ contains one of the original terminals in T. The fact that every face of every minimum-weight inclusionwise minimal multicut dual contains a vertex of T means that the multicut dual has treewidth $O(\sqrt{|T|}) = O(\sqrt{t})$, and then Theorem 11 delivers the required running time.

Consider an optimum solution S' of the instance (G_0, H') and let Q be a component of $G_0 \setminus S'$ that is disjoint from T. We consider two cases. Suppose first that more than $t\pi$ edges of G_0 leave Q. Then we can argue that a solution for (G, H) can be obtained from S' by including every edge of E_π and omitting $\pi + 1$ edges leaving Q, contradicting the optimality of S (this is the point where we exploit that the edges are unweighted). Suppose now that at most $t\pi$ edges leave Q. Then we can use the technique of important cuts [26], [6, Chapter 8] to precisely locate this set of edges (or something equivalent to it). We can show that removing these edges from G makes it $\mathsf{planar} + (\pi - 1)e$, thus we are making progress with this step. This way, after branching into $f(t, \pi)$ possible directions, we can arrive at a planar instance (G_0, H') where every multicut dual has treewidth $O(\sqrt{t})$, making it possible to invoke Theorem 11.

Our last application of Theorem 11 is to give a simple way of proving the algorithmic statements in Theorem 9. More concretely, Theorem 9 2(a) and 3(a) directly follow from the following result. The main idea is that the extra π edges violating planarity can be ignored if the endpoints of these edges are treated as extra terminals. Then the corresponding planar instances can be solved using

Combinatorial results. Theorem 4 considers two cases depending on whether every graph in $\mathcal{H}$ is a trivial pattern or not. It is easy to show that the triangle is a projection of every graph that is not trivial, so part (2) of Theorem 4 can be proved under the assumption that the triangle is the projection of some $H \in \mathcal{H}$.

▶ **Lemma 16.** *Let H be a graph that is not a trivial pattern. Then the triangle is a projection of H.*

The statements of Theorems 5, 6, 9, and 10 contain case distinctions depending on whether $\mathcal{H}$ has bounded distance to extended bicliques. We prove a combinatorial result showing that either this distance is bounded, or $\mathcal{H}$ contains all cliques, or $\mathcal{H}$ contains all complete tripartite graphs.

▶ **Theorem 17.** *If $\mathcal{H}$ is a projection-closed class of graphs, then either (i) it contains K_t for every $t \geq 1$, or (ii) contains $K_{t,t,t}$ for every $t \geq 1$, or (iii) there is a $\mu \geq 0$ such that every $H \in \mathcal{H}$ has distance at most μ to extended bicliques.*

The proof of Theorem 17 considers two cases, depending on the distance of a graph $H \in \mathcal{H}$ to being a cograph. A graph is cograph if it does not contain the 4-vertex path P_4 as an induced subgraph. Suppose that "many" vertices are needed to be deleted to obtain a cograph from H. By a standard connection between hitting and packing, this means that there are many vertex disjoint induced copies of P_4 in H. This implies that H has a projection that contains many vertex-disjoint triangles. We analyze the connections between these triangles using a Ramsey-theoretic argument, and conclude that either a large clique or a large complete tripartite graph can be obtained as a projection. Now consider the case when a cograph H' can be obtained from H by removing a few vertices. We observe two simple alternative characterizations of a cograph being bipartite: (1) it is triangle free or (2) every component is a biclique. If H' contains many vertex disjoint triangles, then we argue as above in the other case. Otherwise, H' (and hence H) can be turned into a bipartite cograph by the deletion of a few vertices. Thus H is a collection of bicliques with a few extra vertices, and then it is easy to argue that one of the conclusions of Theorem 17 holds.

Complexity results. Theorem 17 shows that to obtain our main lower bound, Theorem 6 (3b), it is sufficient to consider two cases: when $\mathcal{H}$ is the class of all cliques and when $\mathcal{H}$ is the class of all tripartite graphs. For the former problem, Theorem 2 almost provides a lower bound, except that the case $t = 3$ needs to be resolved. Then we show how the proof can be adapted to the latter problem.

Let us review the proof of Theorem 2 by Cohen-Addad et al. [4]. This lower bound uses a gadget by Marx [28] for the Multiway Cut problem on planar graphs. Let us briefly recall the properties of this gadget. For some integer Δ, the gadget G_Δ has a planar embedding, where the following distinguished vertices appear on the infinite face (in clockwise order) UL, u_1, $\dots$, $u_{\Delta+1}$, UR, r_1, $\dots$, $r_{\Delta+1}$, DR, $d_{\Delta+1}$, $\dots$, d_1, DL, $\ell_{\Delta+1}$, $\dots$, ℓ_1. Here the four vertices UL, UR, DR, DL are considered terminals. Let $M \subseteq E(G_\Delta)$ be a set of edges such that the removal of these edges disconnects the four terminals from each other. We say that M *represents* the pair $(x, y) \in [\Delta] \times [\Delta]$ if $G_\Delta \setminus M$ has exactly four components and each component contains the terminals in precisely one of the following sets:

- $\{UL, u_1, \dots, u_y, \ell_1, \dots, \ell_x\}$, $\{UR, u_{y+1}, \dots, u_{\Delta+1}, r_1, \dots, r_x\}$,
- $\{DL, d_1, \dots, d_y, \ell_{x+1}, \dots, \ell_{\Delta+1}\}$, $\{DR, d_{y+1}, \dots, d_{\Delta+1}, r_{x+1}, \dots, r_{\Delta+1}\}$.

That is, there is a cut between ℓ_x and ℓ_{x+1} on the left side and there is a cut at precisely the same location, between r_x and r_{x+1}, on the right side, etc.

The gadget G_Δ can be constructed in such a way that every minimum set of edges separating the four terminals is a cut representing some pair (x, y). Moreover, given a set $S \subseteq [\Delta] \times [\Delta]$, we can construct a gadget G_S where every such cut actually represents a pair $(x, y) \in S$, and conversely, for every pair $(x, y) \in S$ there is a minimum cut representing (x, y). This gadget G_S can be used the following way in a hardness proof. Suppose that the solution contains a set of edges that is a minimum cut M separating the four terminals of the gadget; by the properties of the gadget G_S, the cut M represents a pair $(x, y) \in S$. Then, intuitively, the gadget transmits the information "x" between the left and right sides, transmits the information "y" between the upper and lower sides, and at the same time enforces that the two pieces of information are compatible, that is, $(x, y) \in S$ should hold.

Our first observation is that the construction of this gadget G_S, as described by Marx [28], has some stronger property that was not exploited before. Namely, the conclusion about the cut M representing some pair $(x, y) \in S$ holds even if we do not require that M separate all four terminals, but we allow that DL and UR are in the same component of $G_S \setminus M$. In other words, it turns out that every "cheap" cut that satisfies this weaker separation property separates DL and UR as well. We formalize this observation in the following way. A *good cut* of G_S is a subset M of its edges such that the connected components of $G_S \setminus M$ that contain one of UL or DR do not contain any other corners of G_S. We show that the following statement can be obtained by slightly modifying the proof of correctness.

▶ **Lemma 20.** *Given a subset $S \subseteq [\Delta]^2$, the grid gadget G_S can be constructed in time polynomial in Δ, and it has the following properties:*

1. *For every $(x, y) \in S$, the gadget G_S has a cut of weight W^* representing (x, y).*
2. *If a good cut of G_S has weight W^*, then it represents some $(x, y) \in S$.*
3. *Every good cut of G_S has weight at least W^*.*

The proof of Theorem 2 by Cohen-Added et al. [4] considers two regimes: (1) $t = O(g)$ and (2) $t = \Omega(g)$. For the regime $t = O(g)$, they prove the lower bound already for $t = 4$: clearly, this shows the same lower bound for every $t > 4$, as the problem cannot get easier with more terminals. The lower bound is proved by a reduction from binary Constraint Satisfaction Problems (CSP). An instance of a *binary* CSP is a triple (V, D, K), where V is a set of variables, D is a domain of values, K is a set of constraints, each of which is a triple $\langle u, v, R\rangle$ with $u, v \in V$ and $R \subseteq D^2$. A *solution* to a CSP instance (V, D, K) is a function $f : V \to D$ such that $(f(u), f(v)) \in R$ holds for each constraint $\langle u, v, R\rangle$. The *primal graph* of the instance (V, D, K) is the undirected graph with vertex set V where u

and v are adjacent if there is a constraint $\langle u, v, R\rangle$. Cohen-Added et al. [4] proved a lower bound for a special case of binary CSP called 4-Regular Graph Tiling, where the primal graph is a 4-regular bipartite graph and the constraints are of a special form.

Algorithm 1 4-Regular Graph Tiling.

Input: A tuple $(k, \Delta, \Gamma, \{S_v\})$, where
- k and Δ are positive integers,
- Γ is a 4-regular graph (parallel edges allowed) on k vertices in which the edges are labeled U, D, L, R in a way that each vertex is adjacent to exactly one of each label,
- for each vertex v of Γ, we have $S_v \subseteq [\Delta] \times [\Delta]$.

Output: For each vertex v of Γ, a pair $s_v \in S_v$ such that, if $s_v = (i, j)$,
- the first coordinates of $s_{L(v)}$ and $s_{R(v)}$ are both i, and
- the second coordinates of $s_{U(v)}$ and $s_{D(v)}$ are both j,

where $U(v)$, $D(v)$, $L(v)$, and $R(v)$ denote the vertex of Γ that is connected to v via the edge labeled by U, D, L, and R, respectively.

Lower bounds on general binary CSP [19, 27] can be translated to this problem:

▶ **Theorem 21** (Cohen-Addad et al. [4]). *Assuming* ETH*, there exists a universal constant α such that for any fixed integer $k \geq 2$, there is no algorithm that decides all the* 4-Regular Graph Tiling *instances whose underlying graph is bipartite and has at most k vertices, in time $O(\Delta^{\alpha \cdot k/\log k})$.*

The gadgets G_S defined above offer a very convenient reduction from 4-Regular Graph Tiling. We represent each variable v by a gadget G_{S_v} and, for each edge of Γ, we identify the distinguished vertices on two appropriate sides of two gadgets. By our intuitive interpretation of the gadgets, if we consider a minimum cut, then each gadget G_{S_v} represents two pieces of information $(x, y) \in S_v$. Furthermore, if two sides are identified, then the fact that the cut happens on these two sides at the same place implies that the corresponding pieces of information in the two gadgets are equal – precisely as required by the definition of 4-Regular Graph Tiling. Thus we obtain a reduction from 4-Regular Graph Tiling to Multiway Cut with $4|V(\Gamma)| = 2|E(\Gamma)|$ terminals. Observe that each gadget is planar and identification of two sides increases genus by at most one: all the identifications can be done in parallel on a single handle. The final step in the proof of Theorem 21 is to identify every copy of UL in every gadget (and similarly for the copies of UR, DR, and DL). One can observe that these identifications do not change the validity of the reduction and increase genus by $O(|E(\Gamma)|)$. Thus we get a reduction from 4-Regular Graph Tiling to Multiway Cut with 4 terminals (4-Terminal Cut). The stronger property of Lemma 20 allows us to further identify DL and UR without changing the validity of the reduction. Thus we can improve the lower bound from 4 terminals to 3 terminals.

▶ **Theorem 22.**
1. 3-Terminal Cut *on instances with orientable genus $\vec{g}$ is* W[1]*-hard parameterized by $\vec{g}$.*
2. *Assuming the* ETH*, there exists a universal constant α such that for any fixed integer $\vec{g} \geq 0$, there is no algorithm that decides all unweighted* 3-Terminal Cut *instances with orientable genus $\vec{g}$ in time $O(n^{\alpha \cdot (\vec{g}+1)/\log(\vec{g}+2)})$.*

By Lemma 16, the triangle is a projection of every nontrivial pattern. Thus Theorem 22 proves Theorem 6 (2b) and the $t \leq \vec{g}$ case of Theorem 6 (3b).

For the regime $t = \Omega(\vec{g})$, Cohen-Addad et al. [4] present a reduction from CSP with a 4-regular constraint graph having orientable genus $\vec{g}$. The first part of their reduction can be expressed as a lower bound for CSP the following way:

▶ **Theorem 23.** *There is a universal constant α such that for every choice of nonnegative integers s and $\vec{g}$ with $s \geq \vec{g}+1$, there exists a 4-regular multigraph P with $|V(P)| \leq s$ of orientable genus at most $\vec{g}$ and the property that, unless* ETH *fails, there is no algorithm deciding the* CSP *instances (V, D, K) of* CSP *whose primal graph is P and that runs in $O(|D|^{\alpha\sqrt{\vec{g}s+s}/\log(\vec{g}+s)})$.*

The gadgets of Marx [28] reduce CSP to Multiway Cut by introducing one gadget for each variable and a constant number of gadgets for each constraint. This results in an instance of Multiway Cut with the same genus $\vec{g}$ as the CSP instance and $t = O(s)$ terminals. The $t > \vec{g}$ case of Theorem 2 now follows from Theorem 23 and this reduction.

We observe that essentially the same proof can be used to prove a lower bound for Group 3-Terminal Cut if we use Lemma 20.

▶ **Theorem 24.** *Assuming* ETH*, there is a universal constant $\alpha > 0$ such that for any $\vec{g} \geq 0$ and $t \geq 3$, there is no $O(n^{\alpha\sqrt{\vec{g}^2+\vec{g}t+t}/\log(\vec{g}+t)})$ algorithm that solves unweighted* Group 3-Terminal Cut*, even when restricted to instances with orientable genus at most $\vec{g}$ and at most t terminals.*

By Theorem 17, if $\mathcal{H}$ is projection-closed and has unbounded distance to extended bicliques, then $\mathcal{H}$ contains either all cliques or all complete tripartite graphs. Thus in the $t > g$ regime, Theorem 6 (3b) follows from the lower bound for Multiway Cut (Theorem 22) and for Group 3-Terminal Cut (Theorem 24) in the two cases, respectively.

References

1 Pranav Arora, Aritra Banik, Vijay Kumar Paliwal, and Venkatesh Raman. List-coloring – Parameterizing from triviality. *Theor. Comput. Sci.*, 821:102–110, 2020. doi:10.1016/J.TCS.2020.02.022.

2 Leizhen Cai. Parameterized complexity of vertex colouring. *Discret. Appl. Math.*, 127(3):415–429, 2003. doi:10.1016/S0166-218X(02)00242-1.

3 Rajesh Hemant Chitnis, Andreas Emil Feldmann, Mohammad Taghi Hajiaghayi, and Dániel Marx. Tight bounds for planar strongly connected Steiner subgraph with fixed number of terminals (and extensions). *SIAM J. Comput.*, 49(2):318–364, 2020. doi:10.1137/18M122371X.

4 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.

5 Éric Colin de Verdière. Multicuts in planar and bounded-genus graphs with bounded number of terminals. *Algorithmica*, 78:1206–1224, 2017. URL: https://api.semanticscholar.org/CorpusID:686950.

6 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

7 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.

8 Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972. doi:10.1145/321694.321699.

9 Jacob Focke, Florian Hörsch, Shaohua Li, and Dániel Marx. Multicut problems in embedded graphs: The dependency of complexity on the demand pattern. *CoRR*, abs/2312.11086, 2023. doi:10.48550/arXiv.2312.11086.

10 Fedor V. Fomin and Petr A. Golovach. Subexponential parameterized algorithms and kernelization on almost chordal graphs. *Algorithmica*, 83(7):2170–2214, 2021. doi:10.1007/S00453-021-00822-X.

11 Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 515–524. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.62.

12 L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi:10.4153/CJM-1956-045-5.

13 Guilherme C. M. Gomes and Vinícius Fernandes dos Santos. On structural parameterizations of the selective coloring problem. In Carlos E. Ferreira, Orlando Lee, and Flávio Keidi Miyazawa, editors, *Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium, LAGOS 2021, Online Event / São Paulo, Brazil, May 2021*, volume 195 of *Procedia Computer Science*, pages 77–85. Elsevier, 2021. doi:10.1016/J.PROCS.2021.11.013.

14 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In Rodney G. Downey, Michael R. Fellows, and Frank K. H. A. Dehne, editors, *Parameterized and Exact Computation, First International Workshop, IWPEC 2004, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3162 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 2004. doi:10.1007/978-3-540-28639-4_15.

15 Gregory Z. Gutin, Diptapriyo Majumdar, Sebastian Ordyniak, and Magnus Wahlström. Parameterized pre-coloring extension and list coloring problems. *SIAM J. Discret. Math.*, 35(1):575–596, 2021. doi:10.1137/20M1323369.

16 T. C. Hu. Multi-commodity network flows. *Operations Research*, 11(3):344–360, 1963.

17 Lars Jaffke and Bart M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. *Discret. Appl. Math.*, 327:33–46, 2023. doi:10.1016/J.DAM.2022.11.011.

18 Bart M. P. Jansen and Hans L. Bodlaender. Vertex cover kernelization revisited - upper and lower bounds for a refined parameter. *Theory Comput. Syst.*, 53(2):263–299, 2013. doi:10.1007/S00224-012-9393-4.

19 Karthik C. S., Dániel Marx, Marcin Pilipczuk, and Uéverton S. Souza. Conditional lower bounds for sparse parameterized 2-CSP: A streamlined proof. *CoRR*, abs/2311.05913, 2023. doi:10.48550/arXiv.2311.05913.

20 Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 771–780. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.53.

21 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.

22 Philip N. Klein and Dániel Marx. A subexponential parameterized algorithm for subset TSP on planar graphs. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1812–1830. SIAM, 2014. doi:10.1137/1.9781611973402.131.

23 Daniel Lokshtanov, Saket Saurabh, and Magnus Wahlström. Subexponential parameterized odd cycle transversal on planar graphs. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 424–434. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.424.

24 Federico Mancini. Minimum fill-in and treewidth of split+ke and split+kv graphs. *Discret. Appl. Math.*, 158(7):747–754, 2010. doi:10.1016/J.DAM.2008.11.006.

25 Dániel Marx. Parameterized coloring problems on chordal graphs. *Theor. Comput. Sci.*, 351(3):407–424, 2006. doi:10.1016/J.TCS.2005.10.008.

26 Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006. doi:10.1016/J.TCS.2005.10.007.

27 Dániel Marx. Can you beat treewidth? *Theory Comput.*, 6(1):85–112, 2010. doi:10.4086/TOC.2010.V006A005.

28 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.

29 Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 474–484. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00052.

30 Dániel Marx and Michal Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. *ACM Trans. Algorithms*, 18(2):13:1–13:64, 2022. doi:10.1145/3483425.

31 Bojan Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Discret. Math.*, 12(1):6–26, 1999. doi:10.1137/S089548019529248X.

32 Jesper Nederlof. Detecting and counting small patterns in planar graphs in subexponential parameterized time. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1293–1306. ACM, 2020. doi:10.1145/3357713.3384261.

33 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.

34 Yasuhiko Takenaga and Kenichi Higashide. Vertex coloring of comparability+ke and -ke graphs. In Fedor V. Fomin, editor, *Graph-Theoretic Concepts in Computer Science, 32nd International Workshop, WG 2006, Bergen, Norway, June 22-24, 2006, Revised Papers*, volume 4271 of *Lecture Notes in Computer Science*, pages 102–112. Springer, 2006. doi:10.1007/11917496_10.

35 Jan van den Brand, Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, Sushant Sachdeva, and Aaron Sidford. A deterministic almost-linear time algorithm for minimum-cost flow. *CoRR*, abs/2309.16629, 2023. doi:10.48550/arXiv.2309.16629.

36 Jan van den Brand, Yu Gao, Arun Jambulapati, Yin Tat Lee, Yang P. Liu, Richard Peng, and Aaron Sidford. Faster maxflow via improved dynamic spectral vertex sparsifiers. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20–24, 2022*, pages 543–556. ACM, 2022. doi:10.1145/3519935.3520068.