Solving PLANAR k-TERMINAL CUT in $O(n^{c\sqrt{k}})$ time

Philip N. Klein[1]* and Dániel Marx[2]**

[1] Computer Science Department,Brown University,Providence, RI, klein@brown.edu
[2] Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary, dmarx@cs.bme.hu

Abstract. The problem PLANAR k-TERMINAL CUT is as follows: given an undirected planar graph with edge-costs and with k vertices designated as terminals, find a minimum-cost set of edges whose removal pairwise separates the terminals. It was known that the complexity of this problem is $O(n^{2k-4}\log n)$. We show that there is a constant c such that the complexity is $O(n^{c\sqrt{k}})$. This matches a recent lower bound of Marx showing that the $c\sqrt{k}$ term in the exponent is best possible up to the constant c (assuming the Exponential Time Hypothesis).

1 Introduction

MULTIWAY CUT (also called MULTITERMINAL CUT) is a generalization of the classical minimum $s-t$ cut problem: given a undirected graph G with edge-costs and given a subset T of k vertices specified as terminals, the task is to find a minimum-cost set of edges whose deletion pairwise separates the k terminal vertices from each other. The study of the computational complexity of this problem was initiated almost thirty years ago in a widely circulated paper by Dahlhaus, Johnson, Papadimitriou, Seymour, and Yannakakis (eventually published [4, 5]). They showed the problem is NP-hard even for $k = 3$, and they gave a 2-approximation algorithm, which has since been improved [1, 3, 8].

They showed that if k can be arbitrarily large, even the restriction to planar graphs is NP-hard. Therefore, for each positive integer k, they consider the problem PLANAR k-TERMINAL CUT and give an algorithm with a running time of $O((4k)^k n^{2k-1}\log n)$. This bound was since improved by roughly a factor of n^3, to $O(k4^k n^{2k-4}\log n)$, by Hartvigsen [6].[3]

We show that the dependence on k of the exponent of n can be improved from $2k-4$ to $c\sqrt{k}$ for a constant c. In particular, we give an algorithm with running time $d^k \cdot n^{c\sqrt{k}}$ for constants c, d. This shows that the complexity of PLANAR k-TERMINAL CUT is $O(n^{c\sqrt{k}})$. A companion paper [9] shows that this is best possible (up to the particular constant c), assuming the Exponential Time Hypothesis [7].

* Supported in part by National Science Foundation Grant CCF-0964037.

** Research supported by the European Research Council (ERC) grant "PARAMTIGHT: Parameterized complexity and the search for tight complexity results," reference 280152.

[3] The much simpler algorithm of [10] is incorrect; see [2].

Dahlhaus et al. observed that a solution of PLANAR MULTIWAY CUT in the *dual* graph is a planar graph with $O(k)$ branch vertices connected by paths. Thus an algorithm can guess the branch vertices of this planar graph in the dual in time $n^{O(k)}$ and then find min-cost paths between them, subject to constraints about enclosing terminals–constraints that are not readily incorporated into shortest-path computation. Dahlhaus et al. achieve their result by exploiting some structural properties of these paths. Our approach is very different: Our algorithm computes a min Steiner tree on the terminals in the dual graph and cuts the plane open along this tree, thereby forming a cycle on which all the terminals lie, and adds zero-cost edges inside the cycle We prove that there is an optimum solution that uses $O(k)$ zero-cost edges; thus the solution after cutting the tree open can still be described by a planar graph having $O(k)$ vertices and therefore treewidth $O(\sqrt{k})$. Since all the terminals lie on a cycle, the topological constraint that certain paths enclose certain terminals can be completely expressed by requiring that the paths cross the cycle in a certain order. Therefore dynamic programming on a tree decomposition suffices to find the solution in the cut-open graph.

2 Preliminaries

Let G be an undirected graph. For a set X of vertices, $\delta_G(X)$ denotes the set of edges uv such that $u \in X, v \notin X$. Such a set is called a *cut*. A cut is *simple* if both X and $V(G) - X$ induce connected components. For nodes u, v in G, a set S of edges *separates* u and v in G if every u-to-v path includes an edge of S.

Fact 2.1 *S separates u, v iff there is a cut $\delta_G(X)$ such that $u \in X, v \notin X$ and $\delta_G(X) \subseteq S$; moreover, the cut can be chosen to be simple.*

We assume basic knowledge of the definitions of *planar embedded graph*, *faces*, and the planar dual. Let G be a connected planar embedded graph, and let G^* be its dual.

Fact 2.2 *Edge-set S forms a simple cut in G iff S forms a simple cycle in G^*.*

Definition 2.3. *For nodes v_1, v_2 of G, edge-set S* dual-separates *v_1 and v_2 in G if S does not include any edge incident to v_1 or v_2, and, for a face f_1 incident to v_1 and a face f_2 incident to v_2, S separates f_1 and f_2 in the planar dual G^*.*

Lemma 2.4. *If S dual-separates v_1 and v_2 in G then G contains a simple cycle of edges in S that dual-separates v_1 and v_2.*

Proof. For $i = 1, 2$, let f_i be a face of G incident to v_i. By Fact 2.1, S contains the edges of a simple cut in the planar dual G^* that separates f_1 and f_2 in G^*. By Fact 2.2, the edges of this simple cut form a simple cycle in G. □

Definition 2.5. *For edge-set S, let H^* be the subgraph of G^* consisting of S. Each face f of H^* corresponds to a collection X_f of faces of G^* (those embedded in f). We say f* encloses *the faces in X_f. For x a vertex or edge of H^*, we say f encloses x if f encloses all the faces that have x on their boundary. If f is not the infinite face, we consider the faces and vertices enclosed by f to be also enclosed by H^*.*

3 Reducing the problem to the biconnected case

For a pair (G,T) where G is an undirected graph and T is a subset of vertices (the *terminals*), an T*-mcut* (a multiway cut with respect to terminal set T) is a set S of edges such that $G-S$ contains no path between distinct terminals. For disjoint subsets $X,Y \subset T$, we define an (X,Y)*-mcut* to be a set S of edges such that $G-S$ contains no path between vertices of X and no path from X to Y.

For a planar embedded graph G, we say a pair (X,Y) of sets of vertices is *biconnectivity-inducing* in G if every minimum-cost (X,Y)-mcut forms a biconnected subgraph of G^*.

Fix a planar embedded graph G_{in} with positive edge-costs and n vertices. We define two problems:

• *Problem A:* given a set T of k vertices, find a minimum-cost T-mcut.

• *Problem B:* given a pair (X,Y) of vertex-sets where $k=|X|+|Y|$, find an (X,Y)-mcut S such that if (X,Y) is a biconnectivity-inducing pair, then S is guaranteed to be a minimum-cost (X,Y)-mcut.

We show that Problem A can be solved by 2^k calls to an algorithm for Problem B, plus additional $O(3^k)$ time. Let $a(T)$ be the minimum cost of a multiway cut for terminal set T. Let $b(X,Y)$ be a function such that

• if (X,Y) is 2-connectivity-inducing, then $b(X,Y)$ is the minimum cost of an (X,Y)-mcut, and

• otherwise, $b(X,Y)$ is the cost of *some* (X,Y)-mcut.

We use a dynamic program based on the recurrence relation

Lemma 3.1. $a(T) = \min_{\emptyset \neq X \subseteq T} b(X, T-X) + a(T-X)$

Proof. It is trivial that the left-hand side is at most the right hand side: the $(X,T-X)$-mcut and the multiway cut of $T-X$ together gives a multiway cut for T. Our goal is to show that the left-hand side is at least the right-hand side.

We generalize the notion of a multiway cut as follows. Let $X_1,\dots,X_p$ be a partition of T (p is arbitrary). An $(X_1,\dots,X_p)$-mcut is a tuple $(S_1,\dots,S_{p-1})$ of mutually disjoint edge-sets of G such that, for $i=1,\dots,k-1$, $G-S_i$ contains no path between distinct nodes of X_i and no path from a node in X_i to a node in $X_{i+1}\cup X_{i+2}\cup\cdots\cup X_p$. If X_p is singleton then $S_1\cup\cdots\cup S_{p-1}$ is a multiway cut separating all terminals in T.

The cost of a tuple $(S_1,\dots,S_p)$ is the sum of costs of the edges. We say a partition $X_1,\dots,X_p$ is *perfect* if $|X_p|=1$ and the minimum-cost of an $(X_1,\dots,X_p)$-mcut equals $a(T)$. Observe that a perfect partition always exist: in particular, $(T-\{t\},\{t\})$ is a perfect partition for every $t\in T$.

Among all perfect partitions of T, let $\hat{X}_1,\dots,\hat{X}_p$ be the finest, and let $(\hat{S}_1,\dots,\hat{S}_{p-1})$ be a minimum $(\hat{X}_1,\dots,\hat{X}_p)$-mcut. We claim that $(\hat{X}_1,\hat{X}_2\cup\cdots\cup\hat{X}_p)$ is 2-connectivity-inducing. Indeed, if $(\hat{X}_1,\hat{X}_2\cup\cdots\cup\hat{X}_p)$ were not 2-connectivity-inducing—if there were a minimum-cost solution S that was not 2-connected in the dual—the partition $\hat{X}_1,\dots,\hat{X}_p$ could be refined by breaking $\hat{X}_i$ into two parts according to the 2-connected components of S in the dual.

As $(\hat{S}_1,\dots,\hat{S}_{p-1})$ has cost $a(T)$, we have that $a(T)$ is at least the sum of the cost of an $(\hat{X}_1,\hat{X}_2\cup\cdots\cup\hat{X}_p)$-mcut and the cost of a multiway cut for $\hat{X}_2\cup\cdots\cup\hat{X}_p$. By the claim

Fig. 1. Illustrates the reduction. The lines are the edges in the planar dual of a minimum-cost $(\hat{X},\hat{Y})$-mcut. The disks represent terminals. The thin lines represent $\hat{S}$, and the small disks are the terminals enclosed by $\hat{S}$.

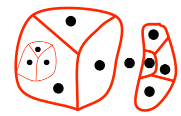
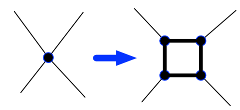


Fig. 2. Each terminal is replaced by a cycle. The size of the cycle is the original degree of the terminal, and the the edges forming the cycle all have cost M.



in the previous paragraph, $(\hat{X}_1, \hat{X}_2 \cup \cdots \cup \hat{X}_p)$ is 2-connectivity-inducing, thus the first term is at least $b(\hat{X}_1, \hat{X}_2 \cup \cdots \cup \hat{X}_p)$. The second term is at least $a(\hat{X}_2 \cup \cdots \cup \hat{X}_p)$. Thus with the choice $X = \hat{X}_1$ shows that the left-hand side is at least the right-hand side. $\square$

4 Algorithm for Problem B

Here is pseudocode for the algorithm for Problem B.

Procedure BSOLVE(G_{in}, X, Y):
input: planar graph G_{in}, pair of disjoint terminal sets (X,Y)
output: (X,Y)-mcut that is min-cost if (X,Y) is biconnectivity-inducing.

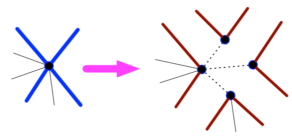
Let M be a number greater than the sum of all costs
0 For each terminal t,
1 replace t by a size-degree(t) cycle of edges of cost M
let t^* (called the *rep* of t) denote the face thus formed
Let $\hat{G}_{\text{in}}$ be the resulting graph and let $\hat{G}^*_{\text{in}}$ denote its planar dual
2 In $\hat{G}^*_{\text{in}}$, find min-cost Steiner tree T^* connecting all terminal reps
3 Replace each edge of T^* with two copies, and replace each node v on T^* with degree(v) copies connected by a star of $d-1$ zero-cost edges
4 Let G_1 denote the planar embedded graph derived in this way from $\hat{G}^*_{\text{in}}$
5 Let $C(G_1)$ be the cycle in G_1 formed by copies of edges of T^*
Label terminal reps by $1,2,\ldots,k$ in clockwise order about $C(G_1)$
6 return the minimum-cost set in
$\{\text{RE}(H,M,G_1) : (H,M) \text{ an } X\text{-valid representative topology}, |M| \le \beta k\}$

Line 1 is illustrated in Fig. 2. Line 3 is illustrated in Figures 3 and 4. In Line 6, β is a constant to be determined.

Line 6 uses the notion of *topology* and the procedure $\text{RE}(H,M,G_1)$. We will presently define this notion. The basic idea underlying the procedure BSOLVE is to enumerate topologies and, for each, find the minimum-cost solution "consistent" with that topology.

• Of course, the procedure cannot enumerate *all* topologies. We will define what it means for topologies to be isomorphic; the procedure will enumerate representatives of distinct isomorphism classes.

Fig. 3. Figure shows part of graph before and after duplicating tree edges (thick edges). Node v on tree is replaced by degree(v) copies connected by a zero-cost star. Graph edges not in tree remain incident to copies of v so as to preserve the embedding.



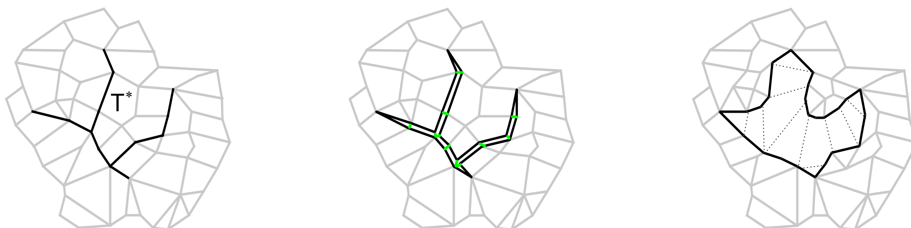


Fig. 4. Cutting along T^* and adding new (dotted) zero-cost edges between copies of the vertices

• Furthermore, we will show it suffices that the procedure consider only representative topologies of *small size*, and that there are not too many such topologies.
• We describe a property, X-*validity*, that captures what a topology must do in order to correspond to an (X,Y)-mcut. The procedure considers only valid representative topologies.
• In Line 6, the procedure RE is invoked on each valid small representative topology. We would like to say that RE finds a minimum-cost topology in G_1 that is isomorphic to the valid representative topology. This is not necessarily true; instead, the procedure finds a valid solution in G_1 whose cost is no greater than the minimum cost of a topology in G_1 isomorphic to the representative.

Definition 4.1. *A* label structure *is a planar embedded graph H containing*
- *a simple cycle $C(H)$ that strictly encloses no nodes, and*
- *a subset of nodes of $C(H)$ labeled $1,2,\ldots,k$ in clockwise order along the cycle (the* terminal reps*, short for representatives).*

Note: *The graph G_1 with the cycle $C(G_1)$ in Lines 4-5 is a label structure.*
Let H be a label structure and let M be a subset of edges. We say M is a feasible solution *for H if no edges of M are incident to labeled nodes. For a subset $X \subset \{1,\ldots,k\}$, we say M is X*-valid *for H if M dual-separates every element of X from every other labeled node in H.*
Let M_1=edges strictly enclosed by $C(H)$ and $M_2 = M - M_1$. We say (H,M) is a topology in *H if, for $i = 1,2$, the edges of M_i form a forest with leaves on $C(H)$. The* size *of (H,M) is $|V(H)|$.*

Definition 4.2. *For a topology (G_1,M_1), where G_1 is the graph obtained in Line 4, the* solution induced in G_{in} *is the set of edges of M_1 that are in G_{in} (including edges of T^* with copies in M_1).*

The definition of dual-separates implies the following lemma.

Lemma 4.3. *An X-valid topology induces an $(X,\{1,\ldots,k\}-X)$-mcut.*

Definition 4.4. *Suppose that, for $i = 1,2$, (G_i, M_i) is a topology. An* isomorphism *between (H_1, M_1) and (H_2, M_2) is a homeomorphism between the subgraph M_1 of H_1 and the subgraph M_2 of H_2 that maps interior edges to interior edges and that preserves the order on the cycle of $\{$endpoints of interior edges$\} \cup \{$labeled nodes$\}$.*

Lemma 4.5. *Isomorphism between topologies preserves X-validity.*

We can bound the number of representative topologies considered in Line 6 by using Catalan numbers:

Lemma 4.6. *The number of isomorphism classes of topologies of size at most s is at most α^s, and representatives of these classes can be enumerated in $O(\alpha^s)$ time, where α is a universal constant.*

This bound depends on the size of the topologies considered; the following theorem, proved in Section 5, states that only small ones need be considered.

Theorem 4.7. *If (X,Y) is biconnectivity-inducing then there is an X-valid topology (G,M) of size at most βk that is isomorphic to a topology in G_1 whose cost is at most that of an optimal (X,Y)-mcut in G_{in}, where β is a universal constant.*

The following theorem is proved in Section 6.

Theorem 4.8. *There is a procedure* $\text{RE}(H, M, G_1)$ *that returns a feasible solution M_1 with the following properties:*
1) If M is X-valid for H then M_1 is X-valid for G_1.
2) If there is a topology (G_1, M_1') isomorphic to (H,M) then M_1 is no more costly than M_1'.
3) The time required is at most $n^{\gamma\sqrt{|V(H)|}}$ for a constants γ.

Finally, putting these results together, we obtain

Theorem 4.9. $\text{BSOLVE}(G_{\text{in}}, X, Y)$ *finds an (X,Y)-mcut in G_{in} that is optimal if (X,Y) is biconnectivity-inducing, and the procedure takes at most $\alpha^{\beta k} n^{c\sqrt{\beta k}}$ time.*

Proof. By Property 1 of Theorem 4.8, BSOLVE returns an X-valid topology of G_1, which by Lemma 4.3 induces an (X,Y)-mcut. We choose the constant β in Line 6 according to Theorem 4.7. Therefore, there exists some small X-valid topology (G,M), among those considered in Line 6, that is isomorphic to a topology (G_1, M_1') in G_1 that induces an optimal (X,Y)-mcut. Therefore, by Property 2 of Theorem 4.8, $\text{RE}(G, M, G_1)$ returns a feasible solution M_1 for G_1 whose cost is at most that of M_1' and therefore at most the optimal cost of an (X,Y)-mcut. The running time is dominated by having to call RE at most $\alpha^{\beta k}$ times (Lemma 4.6), each taking time $n^{\gamma\sqrt{\beta k}}$. □

This theorem plus the reduction to the biconnected case yields our main result, an algorithm for planar k-terminal cut that requires $O(d^k n^{c\sqrt{k}})$ time.

5 Proof of Theorem 4.7

Suppose (X,Y) is biconnectivity-inducing in G_{in}, and let $S \subset E(G_{in})$ be a minimum-cost (X,Y)-cut in G_{in}, breaking ties by minimizing the number of edges not in T^*. Because of the transformation of Line 3 of BSOLVE, the edges of S alone do not dual-separate terminals in G_1, so S is not X-valid for G_1: some zero-cost edges are needed. For a set A of external edges of G_1, define $\text{cr}(A)$ as follows: if A contains edges incident to different copies of the same node of G^*_{in}, include in $\text{cr}(A)$ the internal edges forming a simple path between the different copies. We refer to the edges in $\text{cr}(A)$ as *crossings*.

Lemma 5.1. *For any set A of external edges of G_1, if A induces the solution S in G_{in} then $A \cup \text{cr}(A)$ is X-valid for G_1.*

Among all sets A that induce S, let A_S be one that minimizes $|\text{cr}(A)|$. Without loss of generality, we assume that A_S does not include more than one copy of an edge of S. If G_1 contained a cycle consisting of edges of A_S then G^*_{in} would contain a cycle C consisting of edges of S such that C did not enclose any terminal, so S would not be minimum. Thus A_S is a forest in G_1. A similar argument shows that all the leaves of A_S are endpoints of $\text{cr}(A_S)$. Thus $(G_1, A_S \cup \text{cr}(A_S))$ is a topology in G_1, and it is X-valid by Lemma 5.1. Moreover, since the number of leaves is $\leq 2|\text{cr}(A_S)|$, at most $2|\text{cr}(A_S)|$ nodes have three or more incident edges in A_S. This implies that there is a topology (H,M) isomorphic to $(G_1, A_S \cup \text{cr}(A_S))$ of size at most $3|\text{cr}(A_S)|$. We next show $|\text{cr}(A_S)| \leq 24k$, which implies Theorem 4.7.

Define a *branchpoint* of a graph to be a node of degree three or greater. We refer to the edges of S as *red* edges, and to the subgraph of $\hat{G}^*_{in}$ they form as the *red graph*. We refer to its faces as *red faces*. The *red degree* of a node of $\hat{G}^*_{in}$ is the number of incident red edges. We use *spliced red graph* to refer to the graph obtained from the red graph by splicing out degree-two vertices. By minimality of S, each face of the red graph encloses at least one terminal. Euler's formula then implies $e \leq 3(k-2)$, so the sum of degrees of branchpoints of the red graph is at most $6(k-2)$.

Recall that T^* is a minimum Steiner tree in $\hat{G}^*_{in}$, which we call the *blue graph*. (The red and the blue graphs can share edges.) Each leaf is a terminal rep, so there are k leaves, so the spliced blue graph has at most $2k-3$ edges, so the sum of degrees of branchpoints in the unspliced blue graph is at most $2(2k-3)$.

For a singular red face R, define a *blue ear of R* to be a path B of blue edges such that B connects two nodes on the boundary of a singular red face and each internal node of B is strictly enclosed by R and has blue degree two.

We prove the bound on the number of crossings by a charging scheme, where we charge the crossings to the red branch nodes, blue branch nodes, terminals, and blue ears. We already have a bound of $O(k)$ on the total degree of the branch nodes. The following lemma gives a similar bound on the blue ears.

Lemma 5.2. *The number of blue ears of singular red faces is at most* $14k$.

The proof is illustrated in Figure 5. Let R be a red face, and let R' be the graph obtained from R by including the blue ears of R. Let R'' be the graph obtained from R' by splicing out nodes of blue degree two that are strictly enclosed by R. Consider the

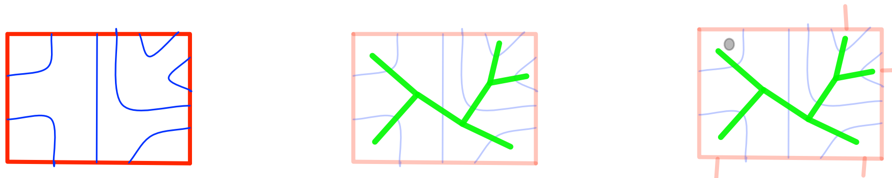


Fig. 5. Proof of Theorem 5.2. On the left is a singular red face (the box) enclosing some blue edges. In the middle is the subgraph of the dual induced by the enclosed faces; it is a tree. As illustrated by the figure on the right, every tree node of degree zero or two is a face that either encloses a terminal or has a red branchpoint on its boundary.

planar dual of R'', and let G_R denote the subgraph of the planar dual consisting of the edges of blue ears. Because every edge of G_R is a cut-edge, we infer that G_R is a tree.

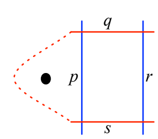
A face of R'' is a *red-blue* face if its boundary consists of a red path and a blue path, and is a *red-blue-red-blue* face if it consists of two red and two blue paths (alternating). The leaves of G_R are red-blue faces in R'', and the degree-two nodes of G_R are red-blue-red-blue faces.

Proposition 5.3. *Every red-blue face either encloses a terminal or has a red branchpoint on its boundary.*

Proof. Suppose PQ is the boundary of a red-blue face, where P is red and Q is blue. If $\mathrm{len}(P) < \mathrm{len}(Q)$ then Q could be replaced in the Steiner tree by P, reducing the length, a contradiction. Therefore $\mathrm{len}(Q) \leq \mathrm{len}(P)$. If PQ does not enclose a terminal and P does not have a branchpoint, replacing P by Q in the optimal solution yields an optimal solution with fewer non-blue edges, a contradiction. □

Proposition 5.4. *The only red-blue-red-blue faces are those that enclose terminals and those that have red branchpoints on their boundary*

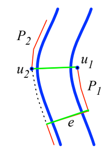
Fig. 6. Illustrates the proof of Lemma 5.4. The horizontal line segments are red, as is the dashed curve, and the vertical line segments are blue. The solid circle represents a terminal in the same red face as the red-blue-red-blue face.



Proof. Suppose F is a red-blue-red-blue face of R'' that does not enclose a terminal and does not have a red branchpoint on its boundary. See Figure 6. The boundary of F is $pqrs$ where p and r are blue and q and s are red, and p divides R'' into a part enclosing F and a part enclosing a terminal.

If $\mathrm{len}(p) \leq \mathrm{len}(q)$, then replacing q with p in the red path yields a solution that is no more expensive but has fewer non-blue edges, a contradiction. Thus $\mathrm{len}(p) > \mathrm{len}(q)$. Similarly $\mathrm{len}(p) > \mathrm{len}(s)$. Removing the path p from the blue graph yields two disconnected components. If the one not containing r contains the intersection of p with s, the graph obtained from the blue graph by replacing p with s is a cheaper solution, a contradiction. The other case is similar. □

Fig. 7. There are two crossings, u_1u_2 and e. P_1 includes no nodes of multiplicity greater than two and no nodes of red degree greater than two. Therefore the path P_1 can be replaced by the dotted line and the two crossings eliminated.



The proof of Lemma 5.2 now follows from the fact that G_R is a tree and from Prop. 5.4 and Prop. 5.3, which bound the number of leaves and degree-two nodes in terms of terminals and red branchpoints.

To complete the proof of the theorem, we now bound the crossings by charging to branchnodes, blue ears, and terminals.

Recall that G_1^* is obtained from $\hat{G}_{in}^*$ by cutting along the edges of T^*, so every edge of T^* is represented in G_1^* by two copies, and every node u of T^* is represented by a number of copies equal to the degree of u in T^*. The *multiplicity* of one such copy is the number of copies, i.e. the degree of u in T^*. If a copy has multiplicity greater than two then u is a branchpoint of the blue graph. The *red degree* of one such copy is defined to be u's red degree in $\hat{G}_{in}^*$ (so here we may count red edges incident to u that are no longer incident to a given copy of u). Let $u_1u_2 \in \mathrm{cr}(A_S)$. In the following, for each case, we assume the previous cases do not hold. By definition of $\mathrm{cr}(A_S)$, there are red edges incident to u_1 and u_2. For $i = 1,2$, let P_i be a maximal path, starting with u_i, of edges in G_1^* that are both red and blue, such that every node of P_i except possibly the last has red degree two and multiplicity two.

Case 1: P_1 or P_2 ends at a branchpoint of the red graph. In this case we charge the crossing to the red branchpoint. The number of crossings charged to such a branchpoint is at most the degree of the branchpoint, so at most $6k$ crossings are charged in this way.

Case 2: P_1 or P_2 ends at a node of multiplicity greater than two. In this case, we charge the crossing to the branchpoint of the blue graph. The number of crossings charged to a branchpoint w by this rule is at most the degree of w in the blue graph. Thus the total number of such crossings is at most $4k$.

Case 3: P_1 or P_2 ends at a node with no incident red edge in G_1^*. Since the red edges form a two-connected subgraph of $\hat{G}_{in}^*$, the last node of P_i has red degree two or more. It follows that in G_1^* some $e \in \mathrm{cr}(A_S)$ is incident to the last node of P_i. However, since every node in P_1 and in P_2 has multiplicity at most two, the configuration is as shown in in Figure 7, and the two crossings can be eliminated, a contradiction.

Case 4: For $i = 1$ and $i = 2$, P_i ends at a node v of red degree two and multiplicity two, but the second red edge incident to v and the second blue edge incident to v differ. Let u be the node of $\hat{G}_{in}^*$ whose copies are u_1 and u_2. Since the red edges form a two-connected subgraph of $\hat{G}_{in}^*$, the neighbors of u in this subgraph are connected in the subgraph by a path Q that avoids u. Let Q' be the cycle obtained from Q by adding the red edges incident to u. (See Figure 8.) For $i = 1,2$, let P_i' be the path obtained from P_i by appending the second red edge incident to the end of P_i.

Because all the nodes of $P_1 \cup P_2$ have red degree two, Q' includes all the edges corresponding to those in $P_1' \cup P_2'$. Let b_i be the blue edge of G_1^* incident to the end of P_i, and let b_i' be the corresponding edge of $\hat{G}_{in}^*$. The cycle Q' shows that b_1' and b_2' are in different faces f_1 and f_2 of the red graph. Because the nodes of $P_1 \cup P_2$ have red

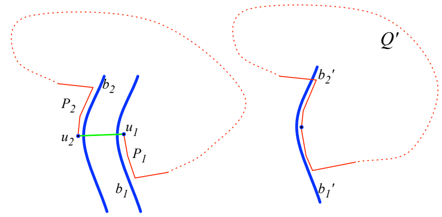


Fig. 8. Case 4. On left, at end of P_i, red path and blue path diverge. Red edge incident to the end of P_i differs from blue edge b_i incident to the end of P_i. Right figure shows $\hat{G}^*_{in}$: a path Q joining the red neighbors of u, forming a cycle Q'. Edges b'_1 and b'_2 are in different but neighboring red faces.

degree two, the edges of $P'_1 \cup P'_2$ belong to the boundaries of f_1 and f_2. The faces f_1 and f_2 cannot both be plural faces, else the edges between them could be removed while maintaining feasibility. Assume without loss of generality that f_2 is a singular face. Let B be a maximal path of blue edges starting with b'_2 such that every node except the last has blue degree two and is strictly internal to f_2.

Subcase a: The last node of B has blue degree one. That last node is a terminal. We charge the crossing to the terminal. There are at most k crossings thus charged.

Subcase b: The last node of B has blue degree greater than two. We charge the crossing to this blue branchpoint. The number of crossings charged to this branchpoint is at most its degree, so the total number of crossings thus charged is at most $4k$.

Subcase c: B forms a path between two nodes on the boundary of f_2. In this case, we charge the crossing to the blue ear. The number of such ears is bounded by Lemma 5.2.

6 Realization

In this section, we prove Theorem 4.8. Given a topology, we try to find a realization of minimum cost in a label structure:

Definition 6.1. *Let (H,M) be a topology of some label structure H, and let G be another label structure. A* realization *of (H,M) in G consists of a mapping $\phi_v : V(M) \to V(G)$ and a mapping $\phi_e : E(M) \to 2^{E(G)}$ such that*

- *ϕ_v preserves the order among $\{$endpoints of interior edges$\} \cup \{$labeled nodes$\}$ on the cycles $C(H)$ and $C(G)$.*
- *For every interior edge $xy \in E(M)$, $\phi_e(xy)$ is an interior edge between $\phi_v(x)$ and $\phi_v(y)$.*
- *For every exterior edge $xy \in E(M)$, $\phi_e(xy)$ is a path of exterior edges between $\phi_v(x)$ and $\phi_v(y)$.*

The cost *of a realization is $\sum_{xy \in E(M)} cost(\phi_e(xy))$.*

Lemma 6.2. *If (H_1,M_1) and (H_2,M_2) are isomorphic topologies and (H_1,M_1) has a realization of cost R in label structure G, then so does (H_2,M_2).*

The following lemma shows that a realization of a valid topology is indeed a solution:

Lemma 6.3. *Let (H,M) be an X-valid topology. Let (ϕ_v,ϕ_e) be a realization of (H,M) in a label structure G having cost R. Then there is an X-valid set $S \subseteq E(G)$ of weight at most R that is X-valid in G.*

Proof. Let S be the union of the edge sets of the path $\phi_e(uv)$ for every edge $uv \in M$. It is clear that the total cost of the edge set S is at most R, the cost of the realization. We claim that S is X-valid in G. Let $i \in X$ be a labeled node and let $j \neq i$ be some other labeled node. By the definition of valid topology and Lemma 2.4, there is a cycle C_1 in M dual-separating i and j. Replacing each interior edge $uv \in C_1$ with the edge $\phi_e(uv)$ and each exterior edge $uv \in C_1$ with the path $\phi_e(uv)$, we can obtain a closed walk C_2 of G. We claim that C_2 dual-separates i and j in G^*.

Let R^1_{ij} and R^2_{ij} be the segment of $C(H)$ (resp., $C(G)$) between labeled nodes i and j in clockwise direction. Let I_1 be the interior edges I_1 with exactly one endpoint on R^1_{ij}. We claim that $|I_1|$ is odd. As C_1 is a simple cycle that dual-separates i and j, there is a dual path Q_1 (i.e., a sequence of faces and edges) in the exterior of H from a face of i to a face of j such that Q_1 contains exactly one edge of C_1. Let R_1 be the set of vertices that can be reached from $R^1_{ij} - \{i,j\}$ on exterior edges without using an edge of Q_1 or going through i or j. By planarity, R_1 does not contain any vertex of the cycle of H outside R^1_{ij}. A simple parity argument shows that the number of edges in the cycle C_1 with exactly one endpoint in R_1 is even. As C_1 does not go through i and j (by the definition of topology), every such edge is either in Q_1 (there is exactly one such edge) or it is an interior edge with exactly one endpoint in R^1_{ij}. Thus there are exactly $|I_1|+1$ such edges and hence $|I_1|$ is odd.

Let $I_2 \subseteq E(G)$ contain those edges of S used by C_2 that have exactly one endpoint in R^2_{ij}. Observe that $|I_1| = |I_2|$: by the definition of realization, the order on the cycle is preserved and hence each edge of I_1 is mapped to a distinct edge of I_2. It also follows that C_2 uses each edge of I_2 only once. Suppose that C_2 does not dual-separate i and j: there is a dual path Q_2 in the exterior of G from a face of i to a face of j. Let R_2 be the set of vertices that can be reached from $R^2_{ij} - \{i,j\}$ on exterior edges of G without using an edge of Q_2 or going through i or j. As C_2 does not go through i and j (by definition of dual-separate) and disjoint from Q_2, only the edges in I_2 have exactly one endpoint in R_2. We have observed that C_2 uses each such edge exactly once and $|I_2| = |I_1|$ is odd, a contradiction. $\square$

In light of Lemma 6.3, all we need is to find minimum-cost realizations of valid topologies. We will use the following embedding result, whose proof uses standard dynamic programming techniques on tree decompositions.

Theorem 6.4. *Let D be a directed graph, U a set of elements, and functions $cv : V(D) \times U \to \mathbb{Z}^+ \cup \{\infty\}$, $ce : V(D) \times V(D) \times U \times U \to \mathbb{Z}^+ \cup \{\infty\}$. In time $|U|^{O(\mathrm{tw(D)})}$, we can find a mapping $\phi : V(D) \to U$ that minimizes*

$$\sum_{v \in V(D)} cv(v, \phi(v)) + \sum_{(u,v) \in E(D)} ce(u, v, \phi(u), \phi(v)).$$

Lemma 6.5. *Given a topology (H,M) and another label structure G, a minimum-cost realization of (H,M) in G can be found in time $|V(G)|^{O(\sqrt{|V(M)|})}$.*

Proof. Let D be the directed graph obtained as an arbitrary orientation of the subgraph of H spanned by M. For every edge $\overrightarrow{xy}$ of D arising from an interior edge of H, we define $ce(x,y,x',y')$ to be 0 if $x'y'$ is an interior edge of G and ∞ otherwise. If $\overrightarrow{xy}$ arises from an exterior edge, then $ce(x,y,x',y')$ is the cost of the shortest path from x' to y' in G containing only exterior edges. We introduce some further directed edges as follows. If x,y are two vertices that are endpoints of interior edges of H such that x is between terminal vertices i and $i+1$ on the cycle and y is the next vertex (in clockwise direction) with this property, then we introduce a directed edge $\overrightarrow{xy}$ and define $ce(x,y,x',y')$ to be 0 if terminal i, vertex x', vertex y', terminal $i+1$ follow each other in this order (in clockwise direction) and ∞ otherwise.

If $x \in V(H)$ (resp., $x' \in V(G)$) is an endpoint of an interior edge of H (resp., G) and it is between i and $i+1$ (resp., i' and $i'+1$) on the cycle in clockwise direction, then we define $cv(x,x') = 0$ if $i = i'$ and $cv(x,x') = \infty$ otherwise. If $x \in V(H)$ is not an endpoint of an interior vertex, then we set $cv(x,x') = 0$ for every $x' \in V(G)$.

Let us use the algorithm of Theorem 6.4 to find a mapping ϕ_v. As D is planar, its treewidth is $O(\sqrt{|V(M)|})$. Therefore, the running time of this step is $|V(G)|^{O(\sqrt{|V(M)|})}$. For every interior edge $xy \in E(M)$, we define $\phi_e(xy)$ to be the interior edge $\phi_v(x)\phi_v(y)$, while if $xy \in E(M)$ is exterior, then we define it to be a shortest path between $\phi_v(x)$ and $\phi_v(y)$ using only the exterior edges of G. It is easy to verify that (ϕ_v, ϕ_e) is a realization of (H,M) in G and its cost is the cost of the mapping ϕ. Furthermore, every realization can be transformed into a mapping with the same cost. Thus the realization obtained this way is indeed a minimum-cost realization. □

To prove Theorem 4.8, the procedure $\textsc{Re}(G,M,G_1)$ uses the algorithm of Lemma 6.5 to find a minimum-cost realization of (G,M) in G_1. By Lemma 6.3, the result is X-valid. The second statement of Theorem 4.8 follows from Lemma 6.2. The running time follows from the statement of Lemma 6.5.

References

1. G. Calinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. In *STOC*, pages 48–52, 1998.
2. K. K. Cheung and K. Harvey. Revisiting a simple algorithm for the planar multiterminal cut problem. *Operations Research Letters*, 38(4):334 – 336, 2010.
3. W. Cunningham and L. Tang. Optimal 3-terminal cuts and linear programming. In *IPCO*, number 1610 in Lecture Notes in Computer Science, pages 114–125, 1999.
4. E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis. The complexity of multiway cuts. In *STOC*, pages 241–251. ACM, 1992.
5. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.
6. D. Hartvigsen. The planar multiterminal cut problem. *Discrete Applied Mathematics*, 85(3):203–222, 1998.
7. R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001.
8. D. R. Karger, P. N. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In *STOC*, pages 668–678, 1999.
9. D. Marx. A tight lower bound for Planar Multiway Cut with fixed number of terminals. To appear in ICALP 2012.
10. W.-C. Yeh. A simple algorithm for the planar multiway cut problem. *J. Algorithms*, 39(1):68–77, 2001.