ELSEVIER
Discrete Applied Mathematics 85 (1998) 203–222
DISCRETE APPLIED MATHEMATICS

The planar multiterminal cut problem

David Hartvigsen*

Department of Management, College of Business Administration, University of Notre Dame, Notre Dame, IN 46556-5641, USA

Received 2 January 1996; received in revised form 10 February 1998; accepted 16 February 1998

Abstract

Let $G=(V,E)$ be a graph with positive edge weights and let $V' \subseteq V$. The min *V'-cut problem* is to find a minimum weight set $E' \subseteq E$ such that no two nodes of V' occur in the same component of $G'=(V,E\backslash E')$. Our main results are two new structural theorems for optimal solutions to the min V'-cut problem when G is planar. The first theorem establishes for the first time a close connection between the planar min V'-cut problem and the well-known "Gomory–Hu" cut collections. The second theorem establishes a connection between the planar min V'-cut problem and a particular matroid. Each theorem results in a simple algorithm for the planar min V'-cut problem. The first algorithm is based upon the most efficient previous algorithm for this problem (due to Dahlhaus et al.) and achieves a lower time complexity.

1

Let $G=(V,E)$ be a graph with positive edge weights and let $V' \subseteq V$. Then $E' \subseteq E$ is called a *V'-cut* if no two nodes of V' occur in the same component of $G'=(V,E\backslash E')$. The *weight* of a V'-cut is the sum of the weights of its edges. The min *V'-cut problem* is to find a minimum weight V'-cut. (This problem has also been referred to as the *k-terminal cut problem*, where $k=|V'|$.) Dahlhaus et al. [4] have proved the following: The min V'-cut problem is NP-hard, even if $|V'| \geqslant 3$ is fixed and all the edge weights are equal to 1 (see also [5]); if only planar graphs are considered, the general problem is still NP-hard; however, if $|V'|$ is fixed and only planar graphs are considered, then there exists a polynomial time algorithm. The history of this problem and several applications are also briefly described in [4]. Additional work using the techniques of polyhedral combinatorics appears in [2, 3].

The min V'-cut problem is a natural generalization of the well-known "min cut" problem where $|V'|=2$. This problem has been extensively studied beginning with the work of Ford and Fulkerson [6]. See [21] for an extensive list of applications and

* E-mail: david.hartvigsen.1@nd.edu.

0166-218X/98/$19.00
PII S0166-218X(98)00036-5

references. Of course, a major difference between the general problem and this special case is that there exists a polynomial algorithm for finding min cuts.

We focus our attention in this paper on the planar min V'-cut problem. We present two main results. Each result is a new structural characterization of optimal solutions to the planar V'-cut problem. We also show how each result leads to a new polynomial time algorithm for this problem, when $|V'|$ is fixed.

The first structural result establishes for the first time a relationship between the min V'-cut problem and a classical result of Gomory and Hu [9]. In particular, we consider an *all V'-pairs min cut collection*, which is a collection of edge sets such that for every pair of nodes $s,t \in V'$, there exists a min $\{s,t\}$-cut in the collection. Gomory and Hu [9] were the first to consider the structure of these cut collections and how to find them. Our result shows that a min V'-cut on a planar graph can essentially be decomposed into a "minimum weight spanning tree structure" and an "all V'-pairs min cut collection". The resulting algorithm (a combination of Kruskal's algorithm and Gomory and Hu's algorithm) produces a min V'-cut for a planar graph in time $O(k\,4^k\, n^{2k-4} \log n)$, where $k = |V'|$ and $n = |V|$. The complexity is better than that of the best previous algorithm for this problem (see [4]), which is $O(k!\,4^k\, n^{2k-1} \log n)$. For example, when $k = 3$, our algorithm has complexity $O(n^2 \log n)$ whereas the previous algorithm has complexity $O(n^5 \log n)$. Our algorithm is based upon the algorithm in [4].

The second structural result shows that an optimal solution to the min V'-cut problem is a minimum weight basis of a particular matroid whose elements are a special collection of paths and cycles in a graph. This connection between the min V'-cut problem and matroids is new. This second result is proved using notions from linear algebra and matroids and is then used to prove the first structural result. Hence, the proof techniques used in this paper are significantly different from those used in [4]. We then show how the paths and cycles described in the above result can be further decomposed into minimum weight paths. This leads to a second algorithm for the min V'-cut algorithm that is based on the greedy algorithm for matroids. This algorithm is also polynomial for $|V'|$ fixed.

Let us finally mention some closely related work. Given a graph $G = (V,E)$ with positive edge weights, the *k-split problem* is to find a minimum weight set of edges $E' \subseteq E$ such that $G' = (V, E/E')$ has at least k components. Goldschmidt and Hochbaum [8] have shown that this problem is NP-hard, but that if k is fixed, it is solvable in polynomial time. The most efficient algorithm for the case that G is planar appears in [10] and has complexity $O(n^{2k-1})$. (This work establishes a relationship between the k-split problem and matroids). For the case that G is planar and $k = 3$, this algorithm can be implemented in time $O(n^3)$. More efficient algorithms for the special case that G is planar and $k = 3$ and all weights are equal appear in Hochbaum and Shmoys [15] and He [14], with complexities $O(n^2)$ and $O(n \log n)$, respectively. Finally, Hassin [13] presents some interesting results that, among other things, give an upper bound on the cardinality of a collection of V'-cuts that contains a min V'-cut for every set of $|V'|$ nodes in a graph. This generalizes an important part of Gomory and Hu's results in [9] concerning the structure of all V'-pairs min cut collections.

The paper is organized as follows. In Section 2 we introduce some basic terminology and state our first structural theorem. In Section 3 we present an algorithm based on the first structural theorem. In Section 4 we introduce some vector space terminology and state our second structural theorem. In Section 5 we present our second algorithm based on the second result. Section 6 contains a proof of the second theorem. Section 7 presents some useful results from [12], which we then use in Section 8 to prove the first theorem.

2

Let $G=(V,E)$ be a graph with positive edge weights. We will frequently assume that G is 2-(node)-connected. This will simplify some definitions, results, and proofs. However, there is no loss of generality in making this assumption since we may always add edges, with very small weights, to make a graph 2-connected without affecting the essential structure of the min weight V'-cuts and without affecting our stated complexities. A *cycle* in G is a minimal subgraph such that every node has degree 2. If G is connected then a *cut* in G is a minimal edge set C such that $G(V,E\backslash C)$ has exactly two components. Note that we consistently use the term *minimal* with respect to set inclusion and the term *minimum* with respect to weights.

Let us refer to a planar graph that has been embedded in the plane as a *plane graph*. A plane graph G divides the plane into maximal open connected sets of points that we refer to as the *regions* of G. Observe that if G is 2-connected, the regions are bounded by cycles of G; we call these cycles *faces*. Let $G^d=(V^d,E)$ denote the "geometric" dual plane graph of G. Observe that if G is 2-connected, then so is G^d. There is a 1–1 correspondence between the regions of G and the nodes of G^d. When G is connected, there is a similar 1–1 correspondence between the nodes of G and the regions of G^d. There is also a 1–1 correspondence between the edge sets of G and G^d, hence we denote both edge sets by E and we let $E'\subseteq E$ denote a subset of edges in both G and G^d. An edge weighting on G induces an edge weighting on G^d. For $E'\subseteq E$, we let $G(E')$ denote the subgraph of G (with no isolated nodes) induced by E'.

We are interested in what a min weight V'-cut in a plane graph corresponds to in its dual. In the simplest case, there is a 1–1 correspondence between a cut in a plane graph and the edge set of a cycle in its dual. The general relationship is characterized by the following definition and proposition.

Let $G=(V,E)$ be a plane graph with positive edge weights and let R' be a subset of its regions. Then $E'\subseteq E$ is called a *R'-separation* if no two regions of R' occur in the same region of $G(E')$. The *weight* of a R'-separation is the sum of the weights of its edges. For an example, see Fig. 1: If E' is the set of bold edges, then E' is a minimum weight R'-separation. We leave the proof of the following proposition to the reader.

Proposition 2.1. *Let $G=(V,E)$ be a connected plane graph with positive edge weights; let $V'\subseteq V$ and let R' be the regions of G^d-corresponding to V'. Then $E'\subseteq E$*

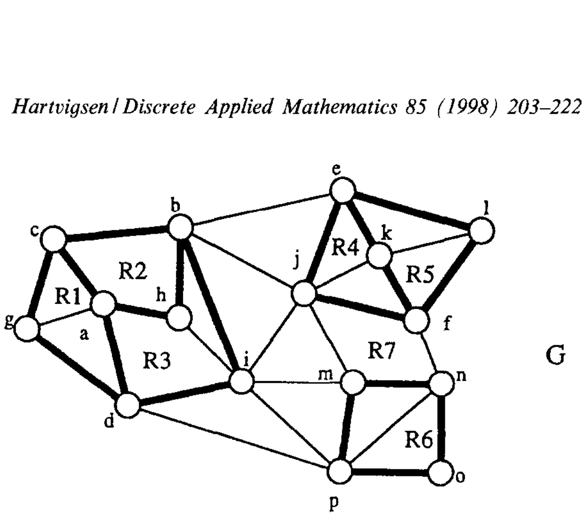

Fig. 1. An optimal topology. All bold edges have weight 1. All non-bold edges have weight 100. $R' = \{R1, \ldots, R7\}$.

is a minimum weight V'-cut of G if and only if E' is a minimum weight R'-separation of G^d.

Hence, we may turn our attention to the problem of finding minimum weight R'-separations in plane graphs, which (for connected graphs) is equivalent to our original problem. We need a few more definitions.

Let $G = (V, E)$ be a graph. A *u–v path* in G is a connected subgraph that contains no cycles and has exactly two degree 1 nodes, u and v, called its *endnodes*. If $U \subseteq \{(u,v)\colon\ u,v \in V,\ u \neq v\}$, then a *$U$-path* is a u–v path such that $(u,v) \in U$. A *topology* T is a pair $\{N, \{N_1, \ldots, N_p\}\}$ where $N \subseteq V$ and $\{N_1, \ldots, N_p\}$ is a partition of N. We let $U(T)$ denote the set of unordered pairs of N such that $(u,v) \in U(T)$ if and only if $u, v \in N_i$ for some $i \in \{1, \ldots, p\}$. We will be interested in $U(T)$-paths.

Consider a plane graph G with positive edge weights and a subset R' of its regions. Let E' be a minimum weight R'-separation in G. The *blocks* of $G(E')$ are its maximal 2-connected subgraphs. Let $G_1, \ldots, G_q$ denote the blocks of $G(E')$ that are not cycles. Let N_i denote the set of nodes with degree three or more in G_i and let $N = \bigcup\{N_i,\ i = 1, \ldots, q\}$. Observe that $|N_i| \geqslant 2$, for $1 \leqslant i \leqslant q$, due to our definition of $G_1, \ldots, G_q$ as non-cyclic blocks.) Such a topology $T = \{N, \{N_1, \ldots, N_q\}\}$ is called an *optimal topology*. Observe that an optimal topology T gives us a unique *decomposition* of $G(E')$ into a union of edge disjoint cycles and paths, namely: the cycles that are blocks of $G(E')$ and the minimal $U(T)$-paths of $G_1, \ldots, G_q$.

Example. See Fig. 1. Again, if E' denotes the set of bold edges, then E' is a minimum weight R'-separation. $G(E')$ contains three blocks, one of which is a cycle. Let G_1 and G_2 denote the non-cycle blocks on the left and right, respectively. Then $N_1 = \{a,b,c,d\}$, $N_2 = \{e,f\}$, $N = N_1 \cup N_2$, and $T = \{N, \{N_1, N_2\}\}$ is an optimal topology. The unique decomposition of $G(E')$ consists of the paths: $(a,h,b), (a,c), (a,d)$ $(b,c), (c,g,d), (b,i,d), (e,j,f), (e,k,f), (e,l,f)$; and the cycle: (m,n,o,p).

Let G be a graph and let $T = \{N, \{N, \ldots, N_p\}\}$ be a topology defined on G. Let $\boldsymbol{P} = \{P_1, \ldots, P_s\}$ be a collection of $U(T)$-paths, where P_i is a u_i–v_i path, and let

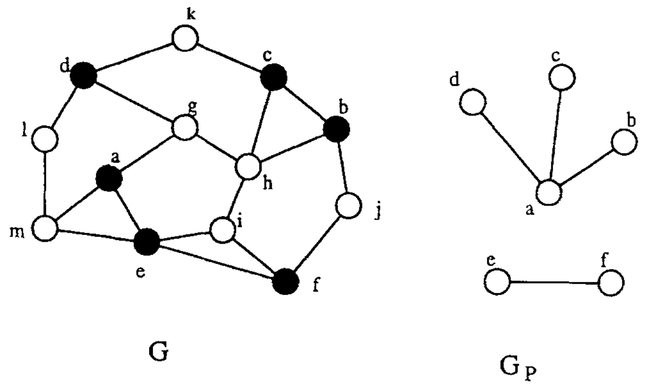

Fig. 2. A spanning T-forest.

$G_p \equiv (N, \{u_i v_i\colon\ i=1,\dots,s\})$. Then $\boldsymbol{P}$ is called a *T-forest* if G_p is a forest. If, in addition, the components of G_p are trees $T_1,\dots,T_p$, where T_i has node set N_i, then $\boldsymbol{P}$ is said to be a *spanning T-forest*. Note that the edges of a T-forest need not induce a forest in the graph G. The *weight* of a T-forest is the sum of the weights of its paths.

Example. See G in Fig. 2. Let $N=\{a,b,c,d,e,f\}$ (=black nodes), $N_1=\{a,b,c,d\}$, $N_2=\{e,f\}$, and $T=\{N,\{N_1,N_2\}\}$. Let $\boldsymbol{P}$ contain the following $U(T)$-paths:

$P_1=(a,g,h,b)$,
$P_2=(a,e,i,h,c)$,
$P_3=(a,g,d)$,
$P_4=(e,i,f)$.

Then G_P, as shown, is a spanning T-forest. Note that the edges in the paths in $\boldsymbol{P}$ do not induce a forest in G.

Our first main theorem follows. The proof appears in Section 8.

Theorem 2.2. *Let $G=(V,E)$ be a 2-connected plane graph with positive edge weights; let $V'\subseteq V$ and let R' be the regions of G^d corresponding to V'; let T be an optimal topology G^d. Let A be the edges in a minimum weight spanning T-forest of G^d and let B be the edges in a minimal all V'-pairs min cut collection of $G(E\backslash A)$. Then $\{A\cup B\}$ is a min V'-cut.*

3

We next present a simple algorithm for the planar min V'-cut problem whose validity follows immediately from Theorem 2.2. The algorithm we present is based upon the algorithm in [4] and similarly depends upon the following proposition.

Proposition 3.1 (Dahlhaus et al. [4]). *Let $G=(V,E)$ be a connected plane graph; let $V' \subseteq V$; let $|V|=n$ and $|V'|=k$. Then, the number of nodes in an optimal topology is at most $2k-4$ and the number of distinct possibilities for an optimal topology is* $O((2n)^{2k-4})$ *or* $O(4^k n^{2k-4})$.

Our algorithm calls upon two subroutines that we now present. The first is basically Kruskal's minimum spanning tree algorithm and the second is Gomory and Hu's algorithm for finding an all V'-pairs min cut collection.

Algorithm. Min weight spanning T-forest

Input: Connected graph $G=(V,E)$ with positive edge weights; topology $T=\{N, \{N_1,\dots,N_p\}\}$; a min weight u–v path P_{uv} for each $uv \in E$.

Output: Min weight spanning T-forest $\boldsymbol{P}$

Step 0: Set $S:=\emptyset$.

Step 1: Form the graph $G'=(N,\{uv\colon u,v \in N_i \text{ for some } i=1,\dots,p\})$; for each edge uv in G', assign it the weight of P_{uv}.

Step 2: Let $\boldsymbol{L}'$ be a list of the edges in G' in non-decreasing order by weight.

Step 3: Consider each edge e in $\boldsymbol{L}'$ in this order and do the following:

If $G(S \cup e)$ has fewer components than $G(S)$, set $S:=S \cup e$.

Step 4: Let $\boldsymbol{P}=\{P_{uv}\colon uv \in S\}$.

End.

That this algorithm works follows immediately from the work of Kruskal [18]. Observe that, in general, there are many possible minimum weight spanning T-forests $\boldsymbol{P}$ that this algorithm could find. The one it finds depends on the choice of minimum weight paths in G as well as the ordering $\boldsymbol{L}'$ of edges in G' used in Step 2.

Suppose $G=(V,E)$ is a connected graph with positive edge weights and that C is a cut. C induces a natural partition of V into two sets, say S and T. We call S and T the *shores* of C. G/S is the graph obtained from G by deleting all edges with both endnodes in S and identifying all the nodes in S into one node. Let C_1 and C_2 be two cuts in G and let S_1, T_1 and S_2, T_2 be the respective partitions of V. Then C_1 and C_2 are called *crossing* if each of the sets $S_1 \cap S_2$, $S_1 \cap T_2$, $T_1 \cap S_2$, and $T_1 \cap T_2$ is non-empty. A collection of cuts is called *non-crossing* if each pair in the collection is non-crossing.

Algorithm. Minimal all V'-pairs min cut collection.

Input: A connected graph $G=(V,E)$ with positive edge weights and $V' \subseteq V$, $|V'| \geqslant 2$.

Output: A minimal all V'-pairs min cut collection $\boldsymbol{C}$.

Step 0: Set $\boldsymbol{C}:=\emptyset$ and $\boldsymbol{S}:=\{G\}$. Call the nodes in V' *unmarked.*

Step 1: If every graph in $\boldsymbol{S}$ has exactly one unmarked node, then done. Otherwise, let H be a graph in $\boldsymbol{S}$ with two or more unmarked nodes.

Step 2: Pick two unmarked nodes, say s and t, in H. Find a min s–t cut C in H and set $\boldsymbol{C}:=\boldsymbol{C} \cup \{C\}$.

Step 3: Let S and T be the shores of C in H. Set $\mathbf{S} := \mathbf{S} \backslash H \cup \{H/S, H/T\}$. Call the nodes resulting from the node identifications in G/S and G/T *marked.* (Note that a node identification may be trivial.) Goto Step 1.

End.

Observation 3.2. The above algorithm outputs an all V'-pairs min cut collection that is not only minimal, but also non-crossing. We note that "minimality" alone is all that is required by Theorem 2.2.

That this algorithm works follows immediately from the work of Gomory and Hu [9]. We next present our algorithm for the planar min V'-cut problem.

Algorithm. Planar min V'-cut (1)

Input: A 2-connected plane graph $G = (V, E)$ with positive edge weights; $V' \subseteq V$.

Output: A minimum weight V'-cut.

Step 1: Find G^d.

Step 2: Find a minimum weight u–v path P_{uv} for every pair of nodes in G^d.

Step 3: For each distinct topology $T = \{N, \{N_1, \ldots, N_p\}\}$ in G^d with $|N| \leqslant 2k - 4$, do the following:

Step 3a: Apply Algorithm: Min weight spanning T-forest to G^d using the paths P_{uv}. Let A_T denote the edges in this T-forest.

Step 3b: Apply Algorithm: Minimal all V'-pairs min cut collection to $G' = (V, E \backslash A_T)$. Let B_T denote the edges in this collection.

Step 4: Output the minimum weight set $\{A_T \cup B_T\}$ found in Step 3.

End.

The validity of this algorithm follows immediately from Theorem 2.2.

Proposition 3.3. *Algorithm: Planar* min V'*-cut can be implemented with worst-case time complexity* $O(k4^k n^{2k-3} \log n)$, *where* $k = |V'|$ *and* $k \geqslant 3$. *In particular, Step* 3 *(i.e., the work for a single topology) can be implemented with worst-case time complexity* $O(kn \log n)$.

Proof. Let us say $|V| = n$ and $|V'| = k$.

Step 1 can be implemented in time $O(n)$.

Step 2 can be implemented in time $O(n^2)$ using an algorithm of Frederickson [7].

Consider Step 3a. The graph G' constructed in Algorithm: Min weight spanning T-forest has $O(k)$ nodes and $O(k^2)$ edges. Hence, Kruskal's algorithm applied to G' requires $O(k^2 \log k)$ time (see, e.g., [24]).

It follows immediately from the work of Gomory and Hu that Step 3b can be implemented in the time it takes to find $k - 1$ min cuts in a planar graph. Frederickson [7] has shown that a single min $s - t$ cut in a planar graph can be found in time

$O(n \log n)$. Frederickson's algorithm relies on the algorithm of Reif [23]. Note that if the addition of an edge $s - t$ leaves the graph planar, then finding a min $s - t$ cut reduces to a single minimum weight path problem in the dual. In this case, a min $s - t$ cut can be found in time $O(n)$ using the algorithm of Klein et al. [17].

It follows that each pass through Steps 3a and 3b requires $O(k^2 \log k + kn \log n) = O(kn \log n)$ time. By Proposition 3.1, we make $O(4^k n^{2k-4})$ passes through these steps, hence the overall complexity of this algorithm is $O(k4^k n^{2k-3} \log n)$. $\square$

Observation 3.4. We could have based Algorithm: Min weight spanning T-forest on Prim's [22] more efficient $O(k^2)$ algorithm. However, this would not improve the complexity of Algorithm: Planar min V'-cut. More importantly, we make use of the "greedy" structure of the algorithm in the proof of (6.6).

The previous algorithm for this problem (see [4]) has complexity $O(k!2^{2k-4} n^{2k-1} \log n)$, which is expressed more compactly in [4] as $O((4k)^k n^{2k-1} \log n)$. A significant difference between the previous algorithm and the algorithm above is that Step 3b in the previous algorithm requires a time that is exponential in k (this accounts for the $k!$ and the k^k terms). Theorem 2.2 allows us to perform Step 3b in polynomial time. In fact, the complexity of Step 3 is polynomial if k is not fixed. Another factor that makes the algorithm in this paper more efficient is that the algorithm in [4] requires the initial edge weights to be perturbed. This perturbation simplifies the proofs in [4] by making the min V'-cut unique. However, it adds a factor of n to the overall complexity. We find in this paper that it is not necssary to perturb the weights. Both algorithms also require min cuts to be found. However, in this paper min cuts are found in planar graphs, whereas in [3] min cuts are found in general graphs. Hence, we are able to make use of a more efficient algorithm (see [7]) that exists for finding min cuts in planar graphs.

4

In this section we begin by defining some elementary vector spaces associated with graphs. We then state our second main theorem that establishes a close relationship between these spaces and minimum weight R'-separations (hence min V'-cuts in planar graphs). We will use this theorem to prove Theorem 2.2 in Section 8.

Let $G = (V, E)$ be a graph. To each subgraph $G' = (V', E')$ (with no isolated nodes) of G we associate an incidence vector x, indexed on E, such that $x_e = 1$ if $e \in E'$ and $x_e = 0$ otherwise. The vector space over GF(2) generated by the incidence vectors of cycles is the well-known *cycle space of G*. A set of cycles whose incidence vectors form a basis for the cycle space of G is called *a cycle basis.* (A well-known example of a cycle basis of a 2-connected plane graph is the set of faces of bounded regions. See [20].) We extend this notion as follows. If $U \subseteq \{(u, v)\colon u, v \in V,\ u \neq v\}$, then the vector space over GF(2) generated by the incidence vectors of cycles and U-paths is called the *U-space of G*. A set of cycles and U-paths whose incidence vectors from a basis for the U-space of G is called a *U-basis*. (We remark that these definitions are structured so that the cycle bases and U-bases are special types of bases for their respective spaces; i.e., a basis for the cycle space (U-space) need not consist of cycles (cycles and U-paths); see (6.2).)

Suppose G has edge weights. Let us say that the *weight of a subgraph of G* is the sum of the weights of its edges and the *weight of a set of subgraphs of G* is the sum of the weights of the subgraphs. *A min U-basis (cycle basis) of G* is a U-basis (cycle basis) of minimum weight.

Let $G=(V,E)$ be a graph and let U be a collection of unordered pairs of distinct nodes of V. For simplicity in our language, we will identify the subgraphs of G (with no isolated nodes) with their incidence vectors. For example, we may refer to the U-space of G as if it contains subgraphs of G instead of the incidence vectors of these subgraphs. Similarly, we may say that a subgraph of G is the sum of a set of other subgraphs, instead of saying the incidence vector of a subgraph is the sum over GF(2) of the incidence vectors of a set of other subgraphs.

Let $\boldsymbol{X}$ be a full rank set of vectors from a finite dimensional vector space and let each vector in $\boldsymbol{X}$ have a weight. If $X \subseteq \boldsymbol{X}$ is independent, then an *extension of X* is a set $Y \subseteq \boldsymbol{X}$ such that $X \cup Y$ is a basis for the vector space. If the sum of the weights of the vectors in an extension is a minimum, then it is called a *min weight extension*.

The following greedy algorithm can be used to find min weight extensions.

Algorithm. Min weight extension.

Input: A full rank set $\boldsymbol{X}$ of weighted vectors from a finite dimensional vector space and some independent $X \subseteq \boldsymbol{X}$.

Output: A min weight set $Y \subseteq \boldsymbol{X}$ such that $X \cup Y$ is a basis for the vector space.

Step 0: Set $Y := \emptyset$.

Step 1: Put the set $\boldsymbol{X} \backslash X$ into non-decreasing order by weights.

Step 2: Consider, in order, each member x of the set $\boldsymbol{X} \backslash X$ and do the following: If $X \cup Y \cup \{x\}$ is independent, set $Y := Y \cup \{x\}$.

End.

The validity of this algorithm follows immediately from elementary matroid theory (see, e.g. [1] or [25]). In particular, the sets Y such that $X \cup Y$ are independent form a matroid (obtained from the original matroid on $\boldsymbol{X}$ by "contracting" the elements in X). Hence, finding a min weight extension is equivalent to finding a min weight base for this matroid. It is well known that the greedy algorithm can be used to find min weight bases for matroids.

We next present our second characterization of planar min V'-cuts. It states that there is a close relationship between the decomposition of a minimum weight R'-separation and a special type of minimum weight extension.

Theorem 4.1. *Let $G=(V,E)$ be a 2-connected plane graph with positive edge weights; let R' be a subset of two or more of its regions; let $T=\{N,\{N_1,\dots,N_p\}\}$ be an optimal topology; and let X be the set of faces of G except for those forming the border of a region in R'. Then*

(4.1) *the decomposition of any minimum weight R'-separation with topology T is a minimum weight extension of X to a $U(T)$-basis of G.*

In addition, if we let E' be the edges in (the subgraphs of) any minimum weight extension of X to a $U(T)$-basis of G, then

(4.2) *$G(E')$ is a minimum weight R'-separation of G;*

(4.3) *T is the optimal topology associated with $G(E')$.*

Observation 4.2. Note that (4.3) states that, if $\boldsymbol{S}$ is a minimum weight extension of X to a $U(T)$-basis and E' is the edge set of $\boldsymbol{S}$, then the cycles in $\boldsymbol{S}$ are blocks of $G(E')$ and the $U(T)$-paths are internally node disjoint; that is, the nodes of $G(E')$ that have degree $\geqslant 3$ are precisely N.

Property (4.2) suggests that if one is given an optimal topology T for a 2-connected plane graph, then one can find a corresponding minimum weight R'-separation as follows: apply Algorithm: Min weight extension to a list of all the $U(T)$-paths and cycles in the graph to find a minimum weight extension of X. However this is not a polynomial time procedure since the list may be exponential in length. In the next section we address this problem.

5

In this section we first show that the paths and cycles in a decomposition of any minimum weight R'-separation have a simple structure, related to minimum weight paths. This result is of interest in itself since it provides some additional understanding of the structure of min V'-cuts in planar graphs. We then show how to use this result, together with Algorithm: Min weight extension, to get a new polynomial algorithm for the planar min V'-cut problem, for fixed $|V'|$. Although the complexity of this algorithm appears to be worse than that of our first algorithm, we find the greedy/matroidal form of it to be sufficiently interesting for a quick presentation.

A u–v path P is called *edge-short* if G contains an edge $e=u'v'$, a minimum weight u'–u path, and a minimum weight v'–v path such that these two paths are node disjoint. A cycle C in G is called *edge-short* if G contains a node w, an edge $e=u'v'$, a minimum weight u'–w path, and a minimum weight v'–w path such that the two paths share only the node w.

The following result follows from Theorem 2.3 in [10].

Proposition 5.1. *Let $G=(V,E)$ be a connected plane graph with positive edge weights and let R' be a subset of its regions. If E' is a minimum weight R' separation, then the paths and cycles in the decomposition of $G(E')$ are edge-short.*

This proposition, together with Theorem 4.1, immediately suggests the following algorithm.

Algorithm. Planar min V'-cut (2)

Input: A connected plane graph $G=(V,E)$ with positive edge weights; $V' \subseteq V$.

Output: A minimum weight V'-cut.

Step 1: Find G^d. Let R' denote the region of G^d corresponding to V'. Let X denote the set of incidence vectors of faces of G^d except for those containing regions in R'.

Step 2: Find a minimum weight u–v path P_{uv} for every pair of nodes in G^d.

Step 3: Generate a collection $\boldsymbol{L}$ of edge-short paths and cycles as follows:

Cycles. For each node v and edge $e=xy$, let G' be the subgraph of G^d induced by e and the edges in P_{xv} and P_{yu}. If G' is a cycle, add its incidence vector to $\boldsymbol{L}$.

Paths. For each pair of nodes u, v in G^d and edge $e=xy$, let G_1 be the subgraph of G^d induced by e and the edges in P_{ux} and P_{vy} and let G_2 be the subgraph induced by e and the edges in P_{uy} and P_{vx}. If G_i is a u–v path, add its incidence vector to $\boldsymbol{L}$, for $i=1,2$.

Step 4: For each distinct topology $T=\{N,\{N_1,\ldots,N_p\}\}$ in G^d with $|N| \leqslant 2k-4$, do the following:

Let $\boldsymbol{L}' \subseteq \boldsymbol{L}$ denote the union of the $U(T)$ – paths in $\boldsymbol{L}$ and the cycles in $\boldsymbol{L}$. Apply Algorithm: Min weight extension with $\boldsymbol{X}=X \cup \boldsymbol{L}'$. (Use Gaussian elimination over GF(2) to determine independence.)

Step 5: Output the minimum weight set Y found in Step 4.

End.

It is not immediate that the above algorithm works. The problem is that there may be multiple minimum weight paths between some pairs of nodes; hence $\boldsymbol{L}'$ need not contain *all* the edge-short cycles and $U(T)$-paths. One way around this problem is to slightly "perturb" the edge weights and to thereby insure that the shortest paths and the min V'-cut are unique. However, this is not necessary, that is, the algorithm works, as stated, starting from any collection of minimum weight paths generated in Step 2. The proof of this is nontrivial, but is essentially the same as the proof of Theorem 2.6 in [10].

Because the worse case time complexity of this algorithm appears to be worse than that of Algorithm: Planar min V'-cut (1), we present only the following analysis.

Proposition 5.2. *Algorithm*: *Planar* min *V'-cut* (2) *can be implemented with worst case time complexity that is polynomial for fixed* $|V'|$.

Proof. Clearly, Steps 1–3 are polynomial in n. By Proposition 3.1, Step 4 is repeated $O(4^k n^{2k-4})$ times. Each Gaussian elimination in Step 4 is polynomial in n, since $|\boldsymbol{L}'|$ is polynomial in n. The result follows. □

6

This section is devoted to proving Theorem 4.1. Propositions 6.2 and 6.4 are simple generalizations of results in [11]. We begin with Proposition 6.1, which contains three simple and well-known facts about cycle spaces of graphs.

Proposition 6.1. *Let $G=(V,E)$ be a connected graph.*
(6.1) *The dimension of the cycle space of G is $|E|-|V|+1$.*
(6.2) *A subgraph G' of G is in the cycle space of G if and only if G' is the union of edge disjoint cycles of G.*
(6.3) *A subgraph G' of G is in the cycle space of G if and only if every node of G' has even degree.*

For $G=(V,E)$, if $V' \subseteq V$, then the edges of E with exactly one node in V' are called the *coboundary of* V'.

For a plane graph G, a cycle C *separates* two regions R_1 and R_2 of G if R_1 and R_2 are contained in different regions of C.

Proposition 6.2. *Let $G=(V,E)$ be a 2-connected plane graph with edge weights; let R' be a subset of two or more of its regions; and let X be the set of faces of G except for those bounding a region of R'. Let $E' \subseteq E$ be the coboundary of a node set $V' \subseteq V$ and let E_1, E_2 be a non-trivial partition of E'. If there exists a cycle C such that $|E_1 \cap C|$ and $|E_2 \cap C|$ are odd, but no such cycle is in X, then every extension of X to a cycle basis contains such a cycle.*

Proof. Observe that, since E' is the coboundary of a node set, each cycle intersects E' in an even number of edges; hence, each cycle has the property that the number of edges in E_1 and E_2 in the cycle have the same parity. Let us call a cycle *odd* or *even* according to this parity. The result follows from the following observation: no odd cycle can be expressed as a mod-2 sum of even cycles. □

Let C_1 and C_2 be two different faces of a 2-connected plane graph G. A C_1–C_2 *path* is a simple path P with one endnode on C_1, one endnode on C_2, and no other node on C_1 or C_2; if C_1 and C_2 have a common node, then P must be one of these nodes. Let E' be the coboundary of the node set of such a path. Then there exists a cyclic ordering of the edges in E', say, $a,b,x_1,\dots,x_r,c,d,x_{r+1},\dots,x_{r+t}$ where $a,b \in C_1$ and $c,d \in C_2$ (see Fig. 3). Note that it is possible that $b=c$ or $a=d$, however, $a \neq b$ and $c \neq d$. We call $\{b,x_1,\dots,x_r,c\}$, $\{d,x_{r+1},\dots,x_{r+t},a\}$ the (C_1,C_2)-partition of P. We immediately have the following proposition.

Proposition 6.3 (Hartvigsen [10]). *Let G be a 2-connected plane graph. Let C_1 and C_2 be two different faces of G that bound regions R_1 and R_2, respectively, and let P be a C_1–C_2 path. Then, a cycle C separates R_1 and R_2 if C intersects each set in the (C_1,C_2)-partition of P an odd number of times.*

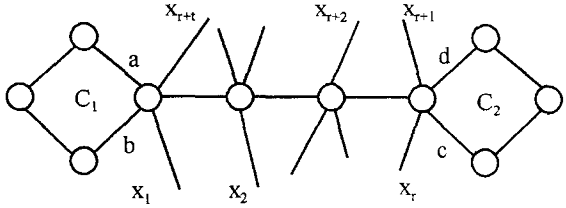

Fig. 3. A $C_1 - C_2$ path and its partition.

Proposition 6.4. *Let G be a 2-connected plane graph with edge weights; let R' be a subset of two or more of its regions; let X be the set of faces of G except for those bounding a region of R'; and let $\boldsymbol{C}$ be an extension of X to a cycle basis. Then for every pair of regions in R', there exists a cycle in $\boldsymbol{C}$ that separates the two regions.*

Proof. The result follows immediately from Propositions 6.2 and 6.3. □

Proposition 6.5. *Let $G=(V,E)$ be a connected graph and let $T=\{N,\{N_1,\dots,N_p\}\}$ be a topology defined on G. Then*

(6.4) *If $\boldsymbol{C}$ is a cycle basis and $\boldsymbol{P}$ is a spanning T-forest, then $\boldsymbol{C}\cup\boldsymbol{P}$ is a $U(T)$-basis.*

(6.5) *The dimension of the $U(T)$-space of G is $|E|-|V|+1+|N|-p$.*

Let X be an independent set of cycles of G. Then

(6.6) For every minimum weight spanning T-forest $\boldsymbol{F}$ there exists a minimum weight extension of X to a $U(T)$-basis that contains $\boldsymbol{F}$.

Proof of (6.4). By definition, $\boldsymbol{C}$ is contained in the $U(T)$-space of G. Let $\boldsymbol{P}=\{P_1,\dots,P_s\}$ where P_i is a u_i–v_i path and let $G\boldsymbol{p}=(N,\{u_iv_i\colon\ i=1,\dots,s\})$.

We first show that $\boldsymbol{C}\cup\boldsymbol{P}$ is independent. We prove this by induction on $|N|$. Clearly, this is true if $|N|=2$, since in this case $|\boldsymbol{P}|=1$ and no path can be expressed as the sum of cycles. So let us assume $|N|>2$ and that the result holds for smaller sets N. Let v be a node in N that has degree 1 in $G\boldsymbol{p}$. Then there exists exactly one path in $\boldsymbol{P}$, say P, with an endnode at v. By inductive hypothesis, $\boldsymbol{C}\cup(\boldsymbol{P}\backslash P)$ is independent in the $U(T')$-space of G, (where T' is the topology induced by removing v from N in T) hence is independent in the $U(T)$-space of G. Since all paths and cycles in $\boldsymbol{C}\cup(\boldsymbol{P}\backslash P)$ have even degree at v, all sums of subsets of $\boldsymbol{C}\cup(\boldsymbol{P}\backslash P)$ must have even degree at v. Since P had odd degree at v, it follows that $\boldsymbol{C}\cup\boldsymbol{P}$ is independent.

To conclude the proof we show that $\boldsymbol{C}\cup\boldsymbol{P}$ spans the entire $U(T)$-space. Let P be a $U(T)$-path not in $\boldsymbol{P}$ with endnodes x and y. Then adding the edge xy to $G\boldsymbol{p}$ creates a unique cycle, call it $C=(Y,Z)$. Let $\boldsymbol{P}'=\{(P_i\colon\ u_iv_i\in\{u_iv_i\colon\ i=1,\dots,s\}\cap Z)\}\cup P$. It is easy to see that each node of G must have even degree in the subgraph corresponding to the sum of the paths in $\boldsymbol{P}'$. Hence, by (6.3), this sum is contained in the cycle space of G. It follows that P can be expressed as the sum of the paths in $\{P_i\colon\ u_iv_i\in\{u_iv_i\colon\ i=1,\dots,s\}\cap Z\}$ plus a set of cycles in $\boldsymbol{C}$. The result follows. □

Proof of (6.5). (6.5) follows immediately from (6.4), the definition of spanning T-forest, and (6.1).

Proof of (6.6). Let $\boldsymbol{F}$ be a min weight spanning T-forest. Let $G' = (N, \{uv\colon u, v \in N_i$ for some $i = 1, \dots, p\})$ and let each edge uv in G' have the weight of a min weight u–v path in G. Then, by definition, G_F is a min weight spanning forest of G'. If P is a u–v path in G, where $\{u, v\} \in U(T)$, let e_P denote the edge uv in G'. Note that, in general, there may be many min weight u–v paths corresponding to the same edge in G'.

Let $\boldsymbol{L}'$ be a list of the edges in G' in non-decreasing order by weight, such that the edges in $\{e_P\colon P \in G_F\}$ are placed as early in the ordering as possible. If we apply Algorithm: Min weight spanning T-forest with the ordering $\boldsymbol{L}'$ in Step 2, and if the paths in $\boldsymbol{F}$ are contained in our collection of min weight paths P_{uv}, then the algorithm outputs $\boldsymbol{F}$.

Let $\boldsymbol{L}$ be a list of all $U(T)$-paths and cycles in G. Order the elements of $\boldsymbol{L}$ in non-decreasing order by weight such that the elements of $\boldsymbol{F}$ are placed as early in the ordering as possible and occur in the same relative order as the corresponding edges in $\boldsymbol{L}'$.

Apply Algorithm: Min weight extension to the ordered set $\boldsymbol{L}$ to find an extension of X. Suppose in Step 2 we consider a path $P \in \boldsymbol{F}$. Let A and B denote the sets of paths and cycles, respectively, already chosen by the algorithm. By our choice of $\boldsymbol{L}$, e_P must connect two components of G_A. To see this, observe that e_P connected two components in our application of Algorithm: Min weight spanning T-forest; if e_P does not connect the same components (by node sets) of G_A in our application of Algorithm: Min weight extension, then there must exist a lighter edge that did so earlier. But this contradicts our placement of $\boldsymbol{L}'$ within $\boldsymbol{L}$. We claim that $A \cup B \cup P$ is independent, hence the algorithm adds P to the minimum weight extension.

To see this let $G_1 = (V_1, E_1)$ be one of the two components of G_A connected by e_P. Let C denote the coboundary of V_1 in G. By our choice of V_1, every path and cycle in A and B has an intersection of even cardinality with C. The sum of any collection of paths and cycles with this property also has this property. But the intersection of P with C has odd cardinality, therefore $A \cup B \cup P$ is independent. □

We use the following classical result from linear algebra to prove our main theorem.

Proposition 6.6. *Let $\boldsymbol{B}$ be a basis for a vector space. If $B \in \boldsymbol{B}$ and $B_1 + \cdots + B_m = B$, then there exists an i in $\{1, \dots, m\}$ such that $\boldsymbol{B} \backslash \{B\} \cup \{B_i\}$ is also a basis for this space.*

Proof of Theorem 4.1. Let us assume, without loss of generality, that R' contains the unbounded region. We begin by proving two claims.

Claim 1. *The decomposition of any minimum weight R'-separation with topology T is an extension of X to a $U(T)$-basis of G.*

Proof of Claim 1. Let E' be a minimum weight R'-separation with topology T. Let $\boldsymbol{D}$ denote the decomposition of $G(E')$. Let $\boldsymbol{P} \subseteq \boldsymbol{D}$ denote a spanning T-forest. Let $\boldsymbol{C}$ denote the collection of cycles in $\boldsymbol{D}$. Observe that every path P in $\boldsymbol{D} \backslash \boldsymbol{P}$ forms a unique cycle, say C_P with the paths in $\boldsymbol{P}$. Let $\boldsymbol{C}'$ denote this collection of cycles. It is easy to see that every cycle in $\boldsymbol{C} \cup \boldsymbol{C}'$ contains in its interior a unique region of R' that is not contained in the interior of any other cycle in $\boldsymbol{C} \cup \boldsymbol{C}'$ and that every region of R', except one, plays such a role. This implies that the number of cycles and paths in $\boldsymbol{D} \backslash \boldsymbol{P}$ is equal to $|R'| - 1$. Since $|\boldsymbol{P}| = |N| - p$, we have that $|\boldsymbol{D}| = |R'| - 1 + |N| - p$. By (6.1) (and the fact that the faces of a plane graph are a cycle basis), $|X| = |E| - |V| + 1 - (|R'| - 1)$. Hence $|\boldsymbol{D} \cup X| \leqslant |R'| - 1 + |N| - p + |E| - |V| + 1 - (|R'| - 1) = |E| - |V| + 1 + |N| - p$, which, by (6.5), is the dimension of the $U(T)$-space of G. Since $\boldsymbol{D}$ contains a spanning T-forest $\boldsymbol{P}$, and since the faces of G are a cycle basis, it suffices to show, by (6.4), that every face of G is in the space generated by $\boldsymbol{D} \cup X$.

Let F be a face of G not in X; i.e., F bounds a region B of R'. Let C be the cycle of $G(E')$ that contains B, and no other region of R', in its interior. Either $C \in \boldsymbol{D}$, or C is the disjoint union of paths in $\boldsymbol{D}$. In either case, C is in the $U(T)$-space. Because B is the only region of R' in the interior of C, F is the sum of C and the faces of X bounding regions interior to C. This proves the claim. □

Proof of Claim 2. *The edges in* (*the subgraphs of*) *any extension of X to a $U(T)$-basis of G contain a R'-separation of G.*

Proof. Let $\boldsymbol{Y}$ be an extension of X to a $U(T)$-basis of G. Because $X \cup \boldsymbol{Y}$ is a $U(T)$-basis, and because the $U(T)$-space contains the cycle space, there must exist a set of graphs Y that can be expressed as sums of graphs in $\boldsymbol{Y}$ and such that $X \cup Y$ is a basis for the cycle space of G. By (6.2), each noncycle in Y is the sum of edge disjoint cycles of G, and by Proposition 6.6 we can substitute for each non-cycle in Y one of these cycles so that $X \cup Y$ is still a basis for the cycle space of G. Thus, we may assume $X \cup Y$ is a cycle basis. Note that all edges that occur in cycles of Y are also edges in graphs in $\boldsymbol{Y}$. Since no cycle in X separates a pair of regions in R', it follows from Proposition 6.4 that for every pair of regions in R', there exists a cycle in Y that separates the pair. It follows that the edges in the graphs of $\boldsymbol{Y}$ contain a R'-separation. This proves the second claim. □

Proof of (4.1). Let $\boldsymbol{D}$ be a minimum weight R'-separation with topology T. Suppose the decomposition of $\boldsymbol{D}$ is not a minimum weight extension. By Claim 1, the decomposition of $\boldsymbol{D}$ is an extension. Thus, if we let $\boldsymbol{C}$ be a minimum weight extension, then the weight of $\boldsymbol{C}$ is less than the weight of $\boldsymbol{D}$. But, by Claim 2, the edges in the subgraphs of $\boldsymbol{C}$ contain a R'-separation, say $\boldsymbol{D}'$. This implies that the weight of $\boldsymbol{D}'$ is less than the weight of $\boldsymbol{D}$, which is a contradiction. □

Proof of (4.2). Let $\boldsymbol{D}$ be a minimum weight extension and let E' be the edges in (the subgraphs of) $\boldsymbol{D}$. Suppose $G(E')$ is not a minimum weight R'-separation. Let $\boldsymbol{R}'$ be a minimum weight R'-separation. The weight of $\boldsymbol{R}'$ is less than the weight of $G(E')$. By (4.1), the decomposition, call it $\boldsymbol{D}'$, of $\boldsymbol{R}'$ is a minimum weight extension; but the weight of $\boldsymbol{D}'$ is less than the weight of $\boldsymbol{D}$, which is a contradiction. □

To prove (4.3), we begin by proving the following claim.

Claim 2. *The paths and cycles in any minimum weight extension of X to a $U(T)$-basis of G are edge disjoint.*

Proof of Claim 3. By assumption, there exists a minimum weight R'-separation, say E'', with topology T and its decomposition clearly consists of edge disjoint paths and cycles. By (4.1), these paths and cycles are a minimum weight extension of X to a $U(T)$-basis of G, hence every minimum weight extension will have this same weight. If there existed a minimum weight extension containing two graphs with a common edge, then the weight of the associated R'-separation (see (4.2)) would be less than that of E'', which is a contradiction. □

Proof of (4.3). Observe that showing (4.3) is equivalent to showing that the number of regions of $G(E')$ equals $|R'|$. Let $\boldsymbol{Y}$ be the decomposition of $G(E')$. Let $\boldsymbol{F} \subseteq \boldsymbol{Y}$ be a spanning T-forest. Consider the operation of sequentially constructing $G(E')$ by adding the paths and cycles in $\boldsymbol{Y} \backslash \boldsymbol{F}$ one at a time to $\boldsymbol{F}$. By Claim 3, all the paths and cycles in $\boldsymbol{Y}$ are edge disjoint, hence the first addition of a path or cycle in this construction will add at least two regions to the induced subgraph of G, and each subsequent addition will add at least one more region. By Proposition 6.5, there are $|R'| - 1$ paths and cycles in $\boldsymbol{Y} \backslash \boldsymbol{F}$, hence $G(E')$ must have at least $|R'|$ regions. If $G(E')$ has more than $|R'|$ regions, then it must contain a region that contains no region in R'. This implies that some edges may be deleted from $G(E')$ to obtain another R'-separation with less weight. This is a contradiction of (4.2). The result follows. □

7

This brief section contains a result from [12] that will be useful in the proof of Theorem 2.2 in the next section. The *weight* of a collection of cuts is the sum of the weights of the cuts in the collection.

Theorem 7.1 (Hartvigsen and Margot [12]). *Let $G = (V, E)$ be a connected graph with positive edge weights, let $V' \subseteq V$; and let $\boldsymbol{C}$ be a collection of non-crossing cuts in G. Then the following are equivalent:*

(7.1) *$\boldsymbol{C}$ is a minimal collection with the following property: for every pair u, v of nodes in V', $\boldsymbol{C}$ contains a* $\min\{u, v\}$*-cut.*

(7.2) *$\boldsymbol{C}$ is a minimum weight collection with the following property: for every pair u, v of nodes in V', $\boldsymbol{C}$ contains a $\{u, v\}$-cut.*

Observe that (7.1) is a "local" condition in terms of min cuts for all pairs in V' whereas (7.2) is a "global" condition with no mention of min cuts. An immediate corollary is the following:

Corollary 7.2. *Let $G=(V,E)$ be a connected graph with positive edge weights and let $V' \subseteq V$. Let $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ be two minimal non-crossing collections of cuts with the following property: for every pair u, v of nodes in V', $\boldsymbol{C}_i$ contains a* $\min\{u,v\}$*-cut. Then $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ have the same weight.*

Observation 7.3. Corollary 7.2 also follows immediately from the work of Gomory and Hu [9].

The following proposition allows us to avoid imposing a non-crossing assumption on B in the statement of Theorem 2.2. It also leads to Corollary 8.1.

Proposition 7.4. *Let $G=(V,E)$ be a connected graph with positive edge weights and let $V' \subseteq V$. Let $\boldsymbol{C}$ be a minimal collection of cuts with the following property: for every pair u, v of nodes in V', $\boldsymbol{C}$ contains a* $\min\{u,v\}$*-cut. Let $\boldsymbol{C}'$ be a minimal non-crossing collection of cuts with the following property: for every pair u, v of nodes in V', $\boldsymbol{C}'$ contains a* $\min\{u,v\}$*-cut. Then the weight of $\boldsymbol{C}$ is less than or equal to the weight of $\boldsymbol{C}'$.*

Proof. Suppose $\boldsymbol{C}$ has one or more pairs that cross and that $\boldsymbol{C}=\{C_1,\dots,C_p\}$ is in non-decreasing order by weight. Suppose that $C_1,\dots,C_i$ are non-crossing, but that C_{i+1} crosses at least one of $C_1,\dots,C_i$. C_{i+1} is a min cut for some pair of nodes x, y that are not separated by $C_1,\dots,C_i$. Apply the standard "uncrossing procedure" (first described by Gomory and Hu [9]; for a very clear and concise exposition see [19, p. 62]) to obtain a cut C'_{i+1} that is a min cut for x, y and that does not cross $C_1,\dots C_i$. Set $\boldsymbol{C} := C \backslash C_{i+1} \cup C'_{i+1}$. Note that there may exist one or more pairs of nodes such that this new $\boldsymbol{C}$ does not contain a cut that separates these pairs. If this is the case, then add one or more min cuts to this new $\boldsymbol{C}$ to obtain a minimal collection of cuts with a min cut for every pair of nodes and whose weight is at least that of the original $\boldsymbol{C}$. Because we considered the cuts in non-decreasing order by weight, $C_1,\dots,C'_{i+1}$ occur in this final $\boldsymbol{C}$ and are non-crossing. Hence, if we continue this process, it will end with a noncrossing collection $\boldsymbol{C}$ whose weight is at least that of the original $\boldsymbol{C}$. By Corollary 7.2, the weight of $\boldsymbol{C}$ at the end of the process is equal to the weight of $\boldsymbol{C}'$. The result follows. □

8

In this final section we prove Theorem 2.2.

Let $G=(V,E)$ be a graph and let $e=uv\in E$. Then *contracting* e is the operation of deleting the edge e from G and identifying the nodes u and v; we let G/e denote the graph so obtained. For $A\subseteq E$, we let G/A denote the graph obtained from G by sequentially contracting the edges in A. It is well known that every sequence of contractions of such a set A results in the same graph, hence this notion is well defined. We observe that in a plane graph, if $G(A)$ is acyclic, then there is a natural 1–1 correspondence between the regions of G and G/A.

We call a collection of cycles in a plane graph *non-crossing* if the corresponding cuts in the dual graph are non-crossing. (Equivalently, a pair of cycles crosses if and only if their bounded regions intersect but neither is contained within the other.)

Proof of Theorem 2.2. Let $\boldsymbol{A}$ be a min weight spanning T-forest with edge set A and let $\boldsymbol{B}$ be a minimal all V'-pairs min cut collection of $G(E\backslash A)$ with edge set B.

Let $\boldsymbol{S}$ be a minimum weight extension of X that contains $\boldsymbol{A}$ and let S denote its edge set. (By (6.6), such an extension exists.) By (4.2), $A\cup S$ is a minimum weight R'-separation. By Observation 4.2, the cycles in $\boldsymbol{S}$ are blocks of $G^d(A\cup S)$ and the paths in $\boldsymbol{A}$ are internally node disjoint. Therefore, the paths and cycles in $\boldsymbol{S}$ become cycles in G^d/A; let $\boldsymbol{B}'$ denote this collection of cycles. Observation 4.2 also implies that the cycles in $\boldsymbol{B}'$ are non-crossing. In fact, the cycles in $\boldsymbol{B}'$ are a minimum weight collection of non-crossing cycles with the following property:

(8.1) For every pair of regions in R' (in G^d/A), $\boldsymbol{B}'$ contains a cycle that separates the pair.

To see this, observe that if there were such a collection of cycles with less weight (non-crossing or not), then the edges in these cycles together with A would yield a R' – separation for G^d with less weight than $A\cup S$.

It is well known that $G(E\backslash A)$ is a dual of G^d/A. Therefore, by the duality of cuts and cycles, $\boldsymbol{B}'$ is a minimum weight collection of non-crossing cuts in $G(E\backslash A)$ with the following property:

(8.2) For every pair of nodes in V' (in $G(E\backslash A)$), $\boldsymbol{B}'$ contains a cut that separates the pair.

By Theorem 7.1, it follows that $\boldsymbol{B}'$ is a minimal non-crossing collection of cuts that contains a min cut for every pair of nodes in V' (in $G(E\backslash A)$). Observe that $\boldsymbol{B}$ is a minimal collection of cuts that contains a min cut for every pair of nodes in V' (in $G(E\backslash A)$). By Proposition 7.4, $\boldsymbol{B}$ has weight less than or equal to $\boldsymbol{B}'$. Observe that $\boldsymbol{B}$ is a collection of cycles in G^d/A that together with $\boldsymbol{A}$ forms a R'-separation in G^d. Hence, the weight of $\boldsymbol{B}$ must, in fact, equal the weight of $\boldsymbol{B}'$ and $\{A\cup B\}$ is a minimum weight R'-separation. □

Corollary 8.1. *Let $G=(V,E)$ be a connected plane graph with positive edge weights; let $V'\subseteq V$ and let R' be the regions of G^d corresponding to V'; let T be an optimal topology in G^d. Let A be the edges in a minimum weight spanning T-forest* *of G^d. Then every minimal all V'-pairs min cut collection of $G(E\backslash A)$ has the same weight.*

The above corollary is a bit surprising because, in general, every minimal all V'-pairs min cut collection (of a connected graph with positive edge weights) has the same weight only if we consider non-crossing collections (see Corollary 7.2). For example, consider a graph that is a cycle with four edges, say e_1, e_2, e_3 and e_4, labeled cyclically. Let each edge have weight 1. Then $\{\{e_1,e_2\},\{e_2,e_3\},\{e_3,e_4\}\}$ and $\{\{e_1,e_3\},\{e_2,e_4\}\}$ are two minimal all pairs min cut collections with different weights. The corollary states that this situation does not occur in the graphs $G(E\backslash A)$ as defined in Theorem 2.2.

References

[1] R.E. Bixby, Matroids and operations research, in: H.S. Greenberg, F.H. Murphy, S.H. Shaw (Eds.), Advanced Techniques in the Practice of Operations Research, North-Holland, New York, 1982, pp. 333–458.

[2] S. Chopra, M.R. Rao, On the multiway cut polyhedron, Networks 21 (1991) 51–89.

[3] W.H. Cunningham, The optimal multiterminal cut problem, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 5, 1991, pp. 105–120.

[4] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P. Seymour, M. Yannakakis, The complexity of multiway cuts, SIAM J. Comput. 23 (1994) 864–894.

[5] M. Fiala, The minimum 3-cut problem: an application of polyhedral combinatorics, honors project, Carleton University, 1986.

[6] L.R. Ford, D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.

[7] G.N. Frederickson, Fast algorithms for shortest paths in planar graphs, with applications, SIAM. J. Comput. 16 (6) (1987) 1004–1022.

[8] O. Goldschmidt, D.S. Hochbaum, A polynomial algorithm for the k-cut problem for fixed k, Math. Oper. Res. 19 (1) (1994) 24–37.

[9] R.E. Gomory, T.C. Hu, Multi-terminal network flows, SIAM J. Appl. Math. 9 (1961) 551–570.

[10] D. Hartvigsen, Minimum path bases, J. Algorithms 15 (1993) 125–142.

[11] D. Hartvigsen, R. Mardon, The all-pairs min cut problem and the min cycle basis problem on planar graphs, SIAM J. Discrete Math. 7 (3) (1994).

[12] D. Hartvigsen, F. Margot, Multiterminal flows and cuts, Oper. Res. Lett. 17 (1995) 201–204.

[13] R. Hassin, Solution bases of multiterminal cut problems, Math. Oper. Res. 13 (4) (1988) 535–542.

[14] X. He, An improved algorithm for the planar 3-cut problem, J. Algorithms 12 (1991) 23–37.

[15] D.S. Hochbaum, D.B. Shmoys, An $O(|V|^2)$ algorithm for the planar 3-cut problem, SIAM J. Algebraic Discrete Meth. 6 (1985) 707–712.

[16] T.C. Hu, Integer Programming and Network Flows, Addison-Wesley, Reading, MA, 1969.

[17] P. Klein, S. Rao, M. Rauch, S. Subramanian, Faster shortest-path algorithms for planar graphs, Proc. of 26th Annual ACM Symp. on Theory of Computing, 1994, pp. 27–37.

[18] J.B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, Proc. Amer. Math. Soc. 7 (1956) 48–50.

[19] L. Lovasz, M.D. Plummer, Matching Theory, North-Holland, Amsterdam, 1986.

[20] S. MacLane, A structural characterization of planar combinatorial graphs, Duke Math. J. 3 (1937) 340–372.

[21] J.C. Picard, M. Queyranne, Selected applications of minimum cuts in networks, INFOR 20 (1982) 394–422.

[22] R.C. Prim, Shortest connection networks and some generalizations, Bell System Technol. J. 36 (1957) 1389–1401.

[23] J.H. Reif, Minimum *s-t* cut of a planar undirected network in $O(n \log^2 n)$ time, SIAM J. Comput. 12 (1) (1983) 71–81.
[24] R.E. Tarjan, Data Structures and Network Algorithms, Soc. for Ind. and App. Math., Philadelphia, 1983.
[25] D.J.A. Welsh, Matroid Theory, Academic Press, London, 1976.