Counting and sampling minimum (s,t)-cuts in weighted planar graphs in polynomial time

Ivona Bezáková

Rochester Institute of Technology, Rochester, NY, USA

Adam J. Friedlander

IBM, Poughkeepsie, NY, USA

Abstract

We give an $O(nd + n \log n)$ algorithm computing the number of minimum (s,t)-cuts in weighted planar graphs, where n is the number of vertices and d is the length of the shortest s-t path in the corresponding unweighted graph. Previously, Ball and Provan gave a polynomial-time algorithm for unweighted planar graphs with both s and t lying on the outer face. Our results hold for all locations of s and t and weighted graphs, and have direct applications in computer vision.

Keywords: Counting and Sampling, Minimum (s,t)-cuts, Weighted planar graphs, Maximum flow, Maximal antichains in partially ordered sets

---

1. Introduction

Graph cuts play an important role in a number of computer vision algorithms. For example, in image segmentation, see, e. g., [4, 5, 6], an image is represented by a graph with pixels as the vertices and an edge connects two pixels if they are neighboring and considered similar; the edge weights capture the similarity measure between the pixels. The underlying graph is often planar, typically grid-like. One of the problems of image segmentation is to separate an object from the background. Many segmentation algorithms

Email addresses: `ib@cs.rit.edu` (Ivona Bezáková), `adam@adamfriedlander.net` (Adam J. Friedlander)

rely on finding a minimum cut between two locations, for example a point from the object and a point from the background (often provided as input from the user). The weight of the minimum cut corresponds to the least energy contour between the locations, viewing the edge weights as the strength of the connection between the respective pixels. Since the distance between the two[1] chosen locations can be arbitrary, we place no restrictions on the relative position of the locations.

Counting problems are closely related to random sampling from the same universe [14]. In image segmentation the current algorithms might suffer from finding an "atypical" cut that does not represent the contour well. Ideally, the user would get the opportunity to choose the best contour out of all possible minimum-cut-based segmentations. However, this might be infeasible because the number of minimum cuts between two chosen positions might be exponential. Having the ability to sample from several minimum cuts provides the user with the option to choose the best of these segmentations while keeping the running time reasonably small.

Minimum cuts are also related to network reliability problems where the vertices are individual computers in a network, edges are connections between computers, and the edge weight captures the probability of connection failure. The number of minimum cuts between two end-points is useful in estimating the probability of disconnecting the network, see, e. g., [1]. Ball and Provan [1] showed that, in case of unweighted (multi)graphs, the problem of counting all minimum (s, t)-cuts is polynomially reducible to the problem of counting all maximal antichains in a poset. (An (s, t)-cut is a set of edges that, if removed, disconnect the vertices s and t, see Section 2 for the formal definition.) Both problems are known to be #P-complete, as shown by the same authors in [18]. Nevertheless, they were able to devise a polynomial-time algorithm counting all minimum (s, t)-cuts in planar graphs, under the assumption that both vertices s and t lie on the outer face. Different variants of the network reliability problem and its connection to minimum cuts were studied in a number of previous works, for example [2, 7, 15, 17, 19].

Our main contribution is an efficient polynomial-time algorithm comput-

[1] If the user is allowed to select several points from the object or the background, then the problem reduces to a minimum cut problem between two new vertices added to the original planar graph: the first is connected to the selected points from the object, the second to the selected points from the background. However, the new graph might be nonplanar – we leave the study of this type of graphs for future work.

ing the number of minimum (s, t)-cuts in all weighted planar graphs and for all pairs of s and t (i. e., we do not impose assumptions on the locations of s and t). (For directed graphs we work under the natural and commonly assumed condition that all vertices are reachable from s and lead to t, see, e. g., [8]. Otherwise, the typical definition of cuts leads to pathological cases, as discussed in Section 4.) We extend the result of Ball and Provan to the case of weighted graphs, showing that this case also polynomially reduces to the problem of counting all maximal (unweighted) antichains in a poset. Our main result, summarized in Theorem 1 below, uses the reduction to devise a polynomial-time algorithm for counting minimum (s, t)-cuts for weighted planar graphs, for all possible pairs of s and t.

Theorem 1. *Let $G = (V, E, w)$ be any (directed) planar graph with edge weights $w : E \to \mathbf{R}^+$. Let $s, t \in V$, $s \neq t$, and assume that all vertices are reachable from s, and t is reachable from every vertex. Then, there is an $O(nd + n \log n)$ algorithm for counting all minimum (s, t)-cuts in G, where $n = |V|$ and d is the smallest number of edges forming a path from s to t in G.*

When both s and t lie on the outer face, it is possible to connect them by an edge, splitting the outer face into two faces. The idea of [1] relies on the fact that the antichain problem can then be solved by counting the number of paths between the two new faces in the dual (directed) planar graph. However, planarity does not allow to add the (s, t) edge for arbitrary locations of s and t. We overcome this problem by showing that we can utilize one of the paths from s to t. Our proof of correctness is significantly more elaborate than the outer face case, yet the underlying algorithm is still relatively simple, as summarized in Algorithm 1.

For completeness, we review results studying the problem of finding one of the minimum (s, t)-cuts in a given weighted planar graph. Building on the work of Itai and Shiloach [12], Reif [20] developed an $O(n \log^2 n)$ divide-and-conquer algorithm for undirected graphs. Janiga and Koubek [13] designed an $O(n \log^2 n \log \log n)$ algorithm for directed planar graphs. The result of Borradaile and Klein [3] yields an $O(n \log n)$ algorithm for all planar graphs. The dual graph plays a central role in all these works.

This paper is organized as follows. We present preliminaries, graph terminology, and notation in Section 2. We state the reduction result in Section 3 and we prove the main result, Theorem 1, in Section 4. Section 5 contains the

Algorithm 1 Counting minimum-(s,t)-cuts in a weighted planar graph G

1: Compute a maximum s-t flow such that the directed flow edges do not form a cycle.
2: Construct $\hat{G}$ by contracting every strongly connected component of the residual graph, let $\hat{s}$ and $\hat{t}$ be the vertices of $\hat{G}$ corresponding to s and t, respectively.
3: Let p be a $\hat{t}$-$\hat{s}$ path in $\hat{G}$. Duplicate all edges of p, creating a new $\hat{t}$-$\hat{s}$ path p'. The new edges are on the left of p when traveling from $\hat{t}$ to $\hat{s}$.
4: Let G' be the graph $\hat{G}$ with the new path p'. Construct a (directed) unweighted dual planar graph G'_d of G', omit edges that cross the edges of p'.
5: For every pair of vertices a, b in G'_d that correspond to faces that share an edge in p', compute the number of all a-b paths in G'_d, using an algorithm for directed acyclic graphs.
6: Return the sum of all numbers computed in step 5.

proofs of the results from Section 3. We present an algorithm for uniformly random sampling of minimum (s,t)-cuts in Section 6.

2. Preliminaries

We denote by $\mathbf{R}$, $\mathbf{R}^+$, and $\mathbf{R}_0^+$ the sets of all real numbers, positive real numbers, and nonnegative real numbers, respectively.

We work with directed graphs throughout the paper. The usual conversion of undirected graphs into directed graphs (for every undirected edge include two directed edges) provides corresponding algorithms for undirected graphs.

Let $G = (V, E, w)$ be a directed graph with positive edge weights $w : E \to \mathbf{R}^+$. Let $s, t \in V, s \neq t$ be two vertices. An *(s,t)-cut of G* is a set of vertices $S \subseteq V$ that contains s but not t. The *value of the cut S* is the sum of the edge weights of the edges going out of the set S, i. e., $\sum_{(u,v):\ u \in S, v \notin S} w(u,v)$. A *minimum (s,t)-cut* has the smallest possible value of all (s,t)-cuts.

Our objective is to *count* the number of all possible minimum (s,t)-cuts of an input graph G.

Minimum cuts are related to network flows. A *flow network* is a directed graph $G = (V, E, c)$ where $c : E \to \mathbf{R}^+$ defines non-negative *edge capacities*. Let $s, t \in V, s \neq t$ be two vertices called the *source* and the *sink*, respectively. A *flow* from s to t is a function $f : E \to \mathbf{R}_0^+$ satisfying the following properties:

- *capacity constraint:* $f(e) \leq c(e)$ for every $e \in E$, and
- *flow conservation:* $\sum_{u:\ (u,v)\in E} f(u,v) = \sum_{u:\ (v,u)\in E} f(v,u)$ for every $v \in V - \{s,t\}$.

An edge e is called a *flow edge in* f if $f(e) > 0$. The *value of the flow* f is the sum of the values of flow edges out of s minus the sum of the flow edges into s, i. e., $\sum_{u:\ (s,u)\in E} f(s,u) - \sum_{u:\ (u,s)\in E} f(u,s)$. A flow is said to be *maximum* if it has the largest possible value among all flows from s to t (we also refer to such flows as s-t flows).

The *residual graph of the flow* f, denoted $G_f = (V, E_f, w_f)$, is a weighted directed graph where E_f contains the following two types of edges:

- for every $e = (u,v) \in E$ with $f(e) < c(e)$, the set E_f contains a *forward edge* $e = (u,v)$ with weight $w_f(e) = c(e) - f(e)$, and
- for every $e = (u,v) \in E$ with $f(e) > 0$, the set E_f contains a *backward edge* $e' = (v,u)$ with weight $w_f(e') = f(e)$.

An *augmenting path* in a residual graph G_f is any path from s to t.

The following is a well-known Maximum-flow Minimum-cut Theorem by Ford and Fulkerson [9].

Theorem 2. *Let $G = (V, E, c)$ be a directed graph with positive edge weights and let $s,t \in V$. Then, the value of the minimum (s,t)-cut in G equals the maximum s-t flow value in the flow network G.*

For more information about network flows, see, e. g., [16]. Most of our terminology and notation follows this reference.

3. Reduction to Forward-cuts

We give a polynomial reduction from the problem of counting minimum (s,t)-cuts in a positively weighted graph to the problem of counting maximal antichains in a poset. A poset can be represented by a directed acyclic graph and an antichain is a set of pairwise unrelated vertices (i. e., no vertex has a predecessor in the set). An antichain is maximal if it cannot be extended by adding another vertex.

Instead of proving our results for antichains, we define a closely related notion that we call forward-cuts. A forward-cut contains the antichain elements and all their predecessors. Moreover, a forward-(a, b)-cut contains the vertex a but not b. The formal definition is summarized below.

Definition 3. Let $G = (V, E)$ be a directed acyclic (multi)graph, and let $a \in V$ be a vertex in G of indegree 0 and $b \in V$ be a vertex in G of outdegree 0. Let S be a subset of the vertices V such that $a \in S$ and $b \notin S$. We say that S is a *forward-(a, b)-cut of G* if there is no edge $(u, v) \in E$ such that $v \in S$ and $u \notin S$.

The reduction result is stated in the following theorem.

Theorem 4. *Let $G = (V, E, c)$ be a (directed) flow network with edge capacities $c : E \to \mathbf{R}^+$. Let $s \in V$ be the source and $t \in V$ be the sink. There exists a directed acyclic graph $\tilde{G} = (\tilde{V}, \tilde{E})$ and vertices $\tilde{s}, \tilde{t} \in \tilde{V}$ such that the number of minimum (s, t)-cuts in G is equal to the number of forward-$(\tilde{t}, \tilde{s})$-cuts in $\tilde{G}$. Moreover, $|\tilde{V}| \leq |V|$, $|\tilde{E}| \leq |E|$, and it is possible to construct $\tilde{G}$ in time $O(|V|^3 + |E|^2)$. Also, if G is planar, then $\tilde{G}$ is planar as well and it can be constructed in time $O(|V| \log |V|)$.*

The proof of the theorem is included in Section 5. In the next section we will deal with graphs where every vertex is reachable from s and leads to t. The following corollary will be used in the proof of Theorem 1. We prove the corollary in Section 5.

Corollary 5. *Suppose that there exists a path from s to every vertex of G and a path from every vertex of G to t. Then, $\tilde{t}$ is the only vertex of indegree 0 and $\tilde{s}$ is the only vertex of outdegree 0 in $\tilde{G}$.*

4. Minimum Cuts in Planar Graphs

The following theorem states that we can count the number of minimum (s, t)-cuts in weighted planar graphs in polynomial-time. We impose a natural condition on the input graphs: we can get to every vertex from s and we can get to t from every vertex. Without this condition, vertices that do not influence connectivity of s and t may artificially increase the number of minimum (s, t)-cuts, as shown on Figure 1.

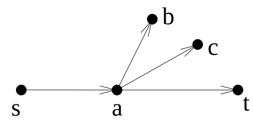

Figure 1: Artificially increasing the number of minimum (s,t)-cuts. All edge capacities are 1. We have several minimum (s,t)-cuts: $\{s\}$, $\{s,b\}$, $\{s,c\}$, $\{s,b,c\}$, and $\{s,a,b,c\}$. However, the "true" minimum (s,t)-cuts are only $\{s\}$ and $\{s,a,b,c\}$.

Theorem 1. *Let $G=(V,E,w)$ be any (directed) planar graph with edge weights $w: E \to \mathbf{R}^+$. Let $s,t \in V$, $s \neq t$, and assume that all vertices are reachable from s, and t is reachable from every vertex. Then, there is an $O(nd + n\log n)$ algorithm for counting all minimum (s,t)-cuts in G, where $n=|V|$ and d is the smallest number of edges forming a path from s to t in G.*

Before we prove the theorem, it will be useful to observe that the number of paths between any two endpoints in a directed acyclic graph can be computed in linear time.

Observation 6. *Let D be a directed acyclic graph and let $a,b \in V(D)$ be two of its vertices. The number of paths from a to b can be counted in time $O(|V(D)|+|E(D)|)$. Moreover, if D is a weighted graph where a path from a to b gets the weight of the product of its edge weights, we can compute the sum of the weights of all paths from a to b in time $O(|V(D)|+|E(D)|)$.*

The statement of the observation follows from sorting the vertices topologically and then, for the starting endpoint a, computing the number of paths from a to every other vertex. This number is simply the sum of the number of paths leading to all neighbors of the current vertex that appear earlier in the topological sort. (In the weighted case we keep track of the sum of the weights of all paths from a to every other vertex.)

Now we are ready to prove the main theorem of this section.

Proof of Theorem 1. Let $\tilde{G}$, $\tilde{s}$, and $\tilde{t}$ be the graph, the source, and the sink from Theorem 4 applied to graph G with edge capacities $c(e)=w(e)$. The theorem states that we need to count the number of forward-$(\tilde{t},\tilde{s})$-cuts in $\tilde{G}$. Suppose $\tilde{G}$ is already embedded in the plane (this can be done in linear time, see, e. g., [11]).

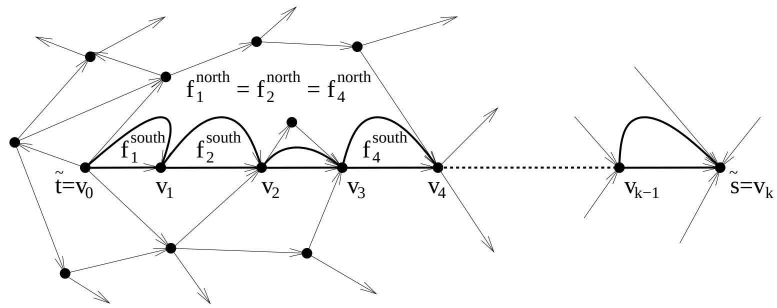

Figure 2: Proof of Theorem 1: The graph G': duplicating the edges on the path p.

Let $p = (\tilde{t} = v_0, v_1, \ldots, v_k = \tilde{s})$ be a (directed) path from $\tilde{t}$ to $\tilde{s}$ in $\tilde{G}$. To simplify our language, let us redraw $\tilde{G}$ so that the path p goes horizontally from left to right. We duplicate every edge of the path, drawing the duplicate edges just above the original edges, see Figure 2. We refer to this new (multi)graph by G' and we use e_i to denote the duplicate edge between vertices v_{i-1} and v_i. Let p' be the path formed by edges $e_1, e_2, \ldots, e_k$. It follows from Corollary 5 that G' is a planar directed acyclic graph with $\tilde{t}$ and $\tilde{s}$ being the only vertices of in- and out-degree 0, respectively.

Every original face bordered by an edge in p' on the south got split into two or more faces in G'. More precisely, the face got split into one or more "south" faces and exactly one "north" face. The "south" faces are in bijection with the e_i edges and we refer to the "south" face corresponding to edge e_i by f_i^{south}. There could be several e_i edges bordering the same "north" face. We refer to the "north" face above the edge e_i by f_i^{north}.

Next we construct a dual graph G'_d and its planar embedding as follows. The faces of G' become the vertices of G'_d and the edges will connect neighboring faces (with one exception, see below). For two neighboring faces f'_1 and f'_2 that share an edge $e' = (u'_1, u'_2)$, there is an edge from f'_1 to f'_2 in G'_d, drawn starting in f'_1, cutting across e', and ending in f'_2, if both of the following conditions are satisfied:

- edge e' is not on the path p',
- if G' is redrawn so that e' is vertical with u'_1 being the bottom end-point, then f'_1 is on the left of e' and f'_2 is on the right.

A possible G' and its dual G'_d are shown on Figure 3. Notice that G'_d is a

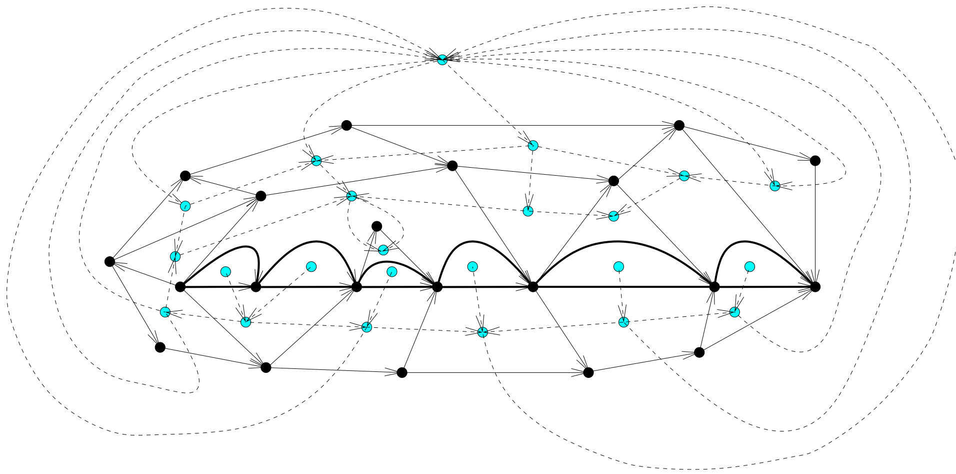

Figure 3: Proof of Theorem 1: Graphs G' (black vertices and solid edges) and G'_d (grey vertices and dashed edges).

planar directed graph that allows multiple edges in case when two faces of G' neighbor in more than one edge.

Claim 1: G'_d is acyclic.

Proof: Suppose that G'_d contains a cycle going through faces $x_1, x_2, \ldots, x_z$ (for convenience let $x_{z+1} = x_1$), see Figure 4. By the definition of edges in G'_d, for every pair of faces x_i, x_{i+1} (in G') there is an edge in G' shared by both faces that "cuts" across the dual edge (x_i, x_{i+1}). Let y_i be the starting endpoint of this edge. Let X be the region defined by the cycle $x_1, x_2, \ldots, x_z$ in G'_d that contains y_1. By the definition of G'_d, besides the edges with endpoints y_i, there are no other edges in G' crossing through the border of X. Moreover, all y_i's lie inside X. Since G' is acyclic, following the predecessors of the y_i's, we must get to a vertex of indegree 0. Thus, $\tilde{t}$, the only vertex of indegree 0, lies inside X. Following the successors of the y_i's, we must get to $\tilde{s}$; therefore, $\tilde{s}$ lies outside of X. Then, the path p' needs to cut through the border of X and must go through one of the y_i's. But G'_d does not contain dual edges crossing through the edges of the path p', a contradiction. $\diamondsuit$

Claim 2: Every path in G'_d that starts at one of the "south" faces f_i^{south} and ends at the corresponding "north" face f_i^{north} uniquely corresponds to a forward-$(\tilde{t}, \tilde{s})$-cut in $\tilde{G}$, and, vice versa, every forward-$(\tilde{t}, \tilde{s})$-cut has a corre-

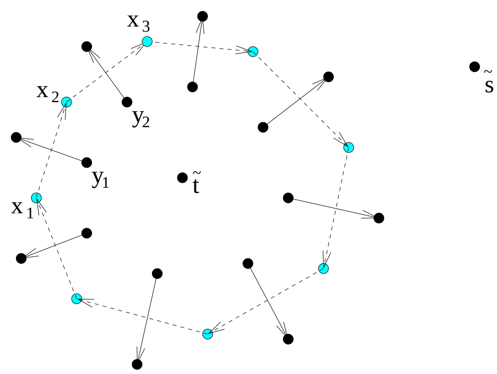

Figure 4: Proof of Claim 1: Region X.

sponding path from a "south" to the corresponding "north" face in G'_d.

Proof: Let q be a path from f_i^{south} to f_i^{north} in G'_d. Let us connect the end-points of q, forming a cycle q'. Notice that the edge connecting the end-points cuts through the path p'. The cycle q' splits the plane into two regions, resulting in a situation similar to Figure 4 (the dashed lines form the cycle q'). Let T be the set of vertices of $\tilde{G}$ that lie in the same region as $\tilde{t}$. The path q cuts through at least one edge of $\tilde{G}$, the edge bordering the face f_i^{south} on the south. The set T includes the left end-point of this edge and the complement of T includes the right end-point. Since $\tilde{s}$ is the only vertex with outdegree 0 and there is a vertex outside T, $\tilde{s}$ must lie outside of T as well. Thus, T is an $(\tilde{t}, \tilde{s})$-cut. By the definition of edges in G'_d, all edges cutting through q start in T and end outside T. By planarity, no other edges connect T with its complement, therefore, T is a forward-$(\tilde{t}, \tilde{s})$-cut. Thus, every "south-north" path defines a forward-$(\tilde{t}, \tilde{s})$-cut.

Vice versa, let T be a forward-$(\tilde{t}, \tilde{s})$-cut. Let (u, v) be such that $u \in T$ and $v \notin T$. We claim that on a face f adjacent to (u, v) there must be exactly one other edge (u', v') such that $u' \in T$ and $v' \notin T$.

First we show that there must be an edge (u', v') such that $u' \in T$ and $v' \notin T$. Let us follow the vertices on the face f, starting with u, going to v, etc. Thus, we start in $u \in T$, then visit $v \notin T$, then there might be other vertices not in T, but eventually we come back to u, a vertex in T. Thus, there must be a pair of consecutive vertices $v' \notin T$ and $u' \in T$ (let u' and v' be the first such encountered pair, besides the edge (u, v), while following the boundary of f). Since T is a forward-cut, the edge between u' and v'

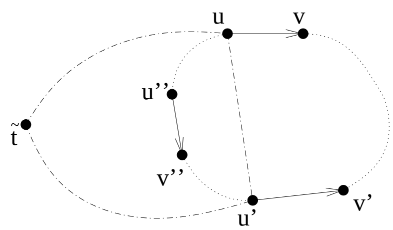

Figure 5: Proof of Claim 2: A region R defined by a $\tilde{t}$-u path, a $\tilde{t}$-u' path, and a connection between u and u'.

must go from u' to v'. (Note that we are allowing multi-edges in $\tilde{G}$. If the face f happens to be defined by only two edges, then $u = u'$ and $v = v'$ but the edges (u, v) and (u', v') are different.)

Next we show that (u, v) and (u', v') are the only two edges on the face f crossing the boundary of T. By contradiction, suppose there is another edge (u'', v'') leading out of T. Suppose we follow the edges on the face f in the cyclic order given by following its boundary, starting with edge (u, v). We will eventually encounter the vertex v', immediately followed by u' (since we defined u', v' as the first encountered pair of vertices such that one is in T and one is not), and later we find (u'', v''), see Figure 5. We know that every vertex, in particular also u and u', is reachable from $\tilde{t}$. Consider a region R defined by a $\tilde{t}$-u path, a $\tilde{t}$-u' path, and a connection between vertices u, u', going through the face f. Notice that $\tilde{t}$ does not lie inside f since f is a face. Additionally, we know that $\tilde{s}$ is reachable from every vertex, including v and v''. We also know that $\tilde{s}$ cannot lie on the paths defining the region since $\tilde{s}$ has outdegree 0. Thus, $\tilde{s}$ must be inside (but not on the paths) or outside (but not on the paths) of R. Then, either v or v'' cannot reach $\tilde{s}$ since exactly one of v and v'' is inside R. We obtained a contradiction with the existence of the third edge (u'', v'').

Therefore, we know that every face containing an edge cutting through T contains exactly two edges leading out of T. Let F be the set of all faces that T cuts through. Then, every face $f \in F$ neighbors two other faces $f^-, f^+ \in F$, where f and f^- share the first edge on f cutting through T, and f and f^+ sharing the second such edge. (In the special case when $|F| = 2$, we have $f^- = f^+$.) Thus, we have $f_1, f_2, \ldots, f_z \in F$ such that for every $i \in \{1, 2, \ldots, z\}$, f_i's only neighbors from F are f_{i-1} and f_{i+1} (let $f_0 = f_z$ and $f_{z+1} = f_1$). In other words, the faces $f_1, f_2, \ldots, f_z \in F$ would form a cycle in the dual graph G'_d, if we included the edges between the "north"

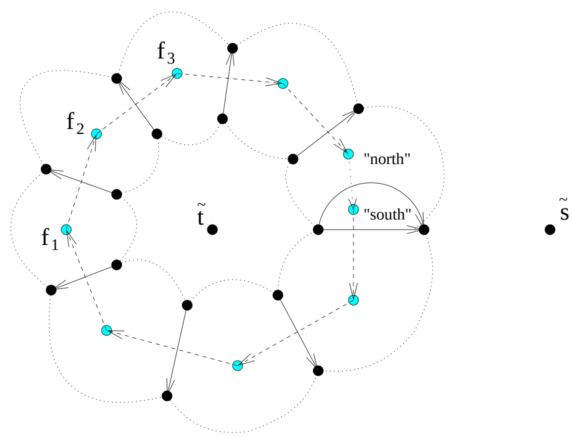

Figure 6: Proof of Claim 2: Faces in F form a cycle in the dual graph G'_d (the dashed edges) with "north"-"south" edges included (the sparsely dotted edge). Vertices of $\tilde{G}$ lying inside this cycle form the forward-$(\tilde{t}, \tilde{s})$-cut T.

and "south" faces. Since $\tilde{t}$ and $\tilde{s}$ are separated by the cycle, the path p' must cut through the cycle. Therefore, there is exactly one "north" and "south" face pair on the cycle and the cycle can be uniquely represented by the corresponding "south" to "north" path in G'_d, see Figure 6.

It remains to argue that the cycle contains all faces from F. If this is not the case, we would have several disjoint cycles of faces from F in G'_d with the added "north"-"south" edges. However, each cycle would need to separate $\tilde{t}$ and $\tilde{s}$, thus the cycles need to be concentric. But then, a $\tilde{t}$-$\tilde{s}$ path cuts through all such cycles and hence it contains at least two edges each leading from a vertex in T to a vertex outside T. This is a contradiction with T being a forward-$(\tilde{t}, \tilde{s})$-cut. $\diamondsuit$

Therefore, by Claim 2, we need to count the number of all paths starting at a "south" face and ending at the corresponding "north" face in G'_d. This can be done, by Observation 6 with $a = f_i^{\text{south}}$ and $b = f_i^{\text{north}}$, in linear time. (Notice that G'_d can be a multi-graph. We then replace multiple edges by a single edge with edge weight equal to the duplicity of the original edges. The weighted path count corresponds to the path count in G'_d.) We need to count the paths for every "south" face f_i^{south} (this happens as many times as is the length of p', i. e., at most n times) and sum over the returned values. The overall running time is $O(|p|(|E|+|V|)) = O(dn)$ since for planar graphs $|E| = O(|V|)$. This running time includes the construction of the path p' and the dual graph G'_d. Accounting for the running time in Theorem 4, we get an overall running time of $O(dn + n \log n)$. $\square$

5. Proof of Theorem 4

In this section we show how to reduce the problem of counting minimum (s,t)-cuts in a weighted directed graph G to the problem of counting forward-$(\tilde{t},\tilde{s})$-cuts in a directed acyclic graph $\tilde{G}$. To simplify the running time estimates, we assume that G, when viewed as an undirected graph, is connected (and, hence, $|V| = O(|E|)$).

We mentioned the connection between minimum cuts and maximum flows. We utilize the connection in our proofs where we work with network flows that are acyclic, as defined below.

Definition 7. Let $G = (V, E, c)$ be a (directed) flow network with edge capacities $c : E \to \mathbf{R}^+$. Let $s \in V$ be the source and $t \in V$ be the sink. We say that an s-t flow $f : E \to \mathbf{R}_0^+$ is *acyclic* if the (directed) graph $F_f = (V, D_f)$ is acyclic, where D_f consists of edges in E that carry positive f-values (formally, $D_f = \{e \in E \mid f(e) > 0\}$). We call the graph F_f the *flow graph* of the flow f.

Notice that the flow graph consists of the backward edges in the corresponding residual graph, reversed. The next claim observes that there exists an acyclic maximum flow in every flow network.

Observation 8. *There exists an acyclic maximum s-t flow in any flow network $G = (V, E, c)$. The acyclic flow can be found in time $O(T(G) + |E|^2)$, where $T(G)$ is the time required by the fastest maximum flow algorithm for G.*

The statement follows by observing that, for a maximum s-t flow $g : E \to \mathbf{R}_0^+$, we can iteratively eliminate cycles from g by decreasing the flow value along each edge on a cycle C by $\min_{e \in C} g(e)$. Thus, in addition to the time $T(G)$, we need at most $|E|$ iterations (every iteration sets the flow through at least one edge to zero), each iteration taking $O(|E|)$ steps (to find a cycle in a directed graph).

Remark 9. *$T(G) = O(|V|^3)$ when using the push-relabel algorithm [10], yielding an $O(|V|^3 + |E|^2)$ algorithm for finding an acyclic maximum s-t flow in any flow network. For planar graphs Borradaile and Klein's [3] maximum flow algorithm returns an acyclic maximum s-t flow in time $O(|V| \log |V|)$.*

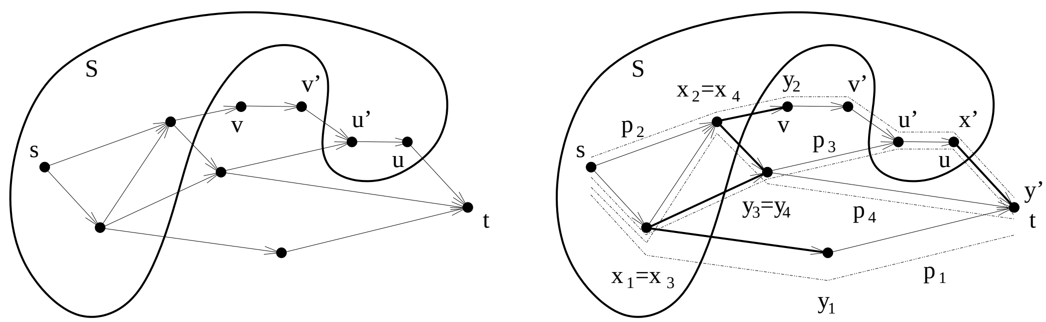

Figure 7: Proof of Observation 11: The left figure shows a graph F_f and an (s,t)-cut S such that there are vertices $u \in S$ and $v \notin S$, and there is a path from v to u. The right figure shows the graph F_f decomposed into paths p_1, p_2, p_3, p_4 and the cut-edges (x_i, y_i) and (x', y') (these edges are highlighted).

In subsequent proofs we rely on the fact that every maximum flow can be obtained from a sequence of augmenting paths that do not use backward edges, as spelled out by the following observation.

Observation 10. *Let $f : E \rightarrow \mathbf{R}_0^+$ be an acyclic s-t flow in a flow network $G = (V, E, c)$ with source s and sink t. Then, f can be decomposed into augmenting paths $p_1, \ldots, p_d$, where $d \leq |E|$.*

Next we state that if a vertex lies in a minimum cut set, then the cut set must contain all of the vertex's predecessors in the flow graph.

Observation 11. *Let $f : E \rightarrow \mathbf{R}_0^+$ be an acyclic maximum s-t flow in a flow network $G = (V, E, c)$ with source s and sink t. Let S be a minimum (s,t)-cut in G. Then, if a vertex u is in S then every vertex v that precedes u in the flow graph F_f must be in S as well.*

Proof. By contradiction, assume that there are two vertices u, v such that $u \in S$, $v \notin S$ and there exists a path from v to u in F_f, see Figure 7. It follows that there is an edge (v', u') in F_f such that $u' \in S$ and $v' \notin S$. By Observation 10, the flow f can be decomposed into d augmenting paths $p_1, \ldots, p_d$ where the path p_i carries flow of value $\phi_i > 0$. At least one of the paths contains the edge (v', u'), let p_j be such a path. For every $i \in \{1, \ldots, d\}$ there exists an edge (x_i, y_i) on the path p_i such that $x_i \in S$, $y_i \notin S$. Moreover, for the path p_j there exists another edge (x', y') besides (x_j, y_j) such that $x' \in S$ and $y' \notin S$.

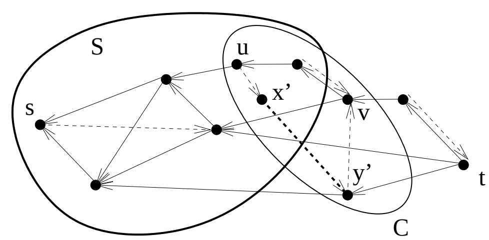

Figure 8: Proof of Observation 12: The figure shows a residual graph G_f and an (s,t)-cut S cutting through a strongly connected component C. Backward edges are depicted by solid arrows and forward edges are dashed. The edge (x',y') is highlighted. For clarity we do not include residual capacities.

The cut value is defined as the sum of the capacities of the edges (x,y) where $x \in S$ and $y \notin S$. Therefore,

$$\begin{aligned}\text{value}(S) &= \sum_{\substack{(x,y)\in E:\\ x\in S, y\notin S}} c(x,y) \geq \sum_{\substack{(x,y)\in E:\\ \exists i:\ x=x_i, y=y_i\\ \text{or } x=x', y=y'}} c(x,y)\\ &\geq \sum_{i=1}^{d} \phi_i + \phi_j = \text{value}(f) + \phi_j.\end{aligned}$$

Since $\phi_j > 0$, the value of the cut S is strictly greater than the value of the flow f. Therefore, by the Max-flow-min-cut Theorem (Theorem 2), S cannot be a minimum (s,t)-cut of G. □

Next we claim that no minimum cut cuts through strongly connected components of a residual graph corresponding to a maximum flow.

Observation 12. *Let $f : E \to \mathbf{R}_0^+$ be an acyclic maximum s-t flow in a flow network $G = (V,E,c)$ with source s and sink t. Let G_f be the corresponding residual graph. If S is a minimum (s,t)-cut in G then, for each strongly connected component C in G_f, either $C \subseteq S$ or $C \cap S = \emptyset$.*

Proof. By contradiction, suppose that there exists a strongly connected component C and a minimum (s,t)-cut S such that there exist vertices $u, v \in C$ such that $u \in S$ and $v \notin S$, see Figure 8. Since u, v belong to the same strongly connected component, there exists a path from u to v in G_f, as well as a path from v to u in G_f. By the definition of a residual graph, G_f contains two types of edges: forward and backward edges where the backward edges are simply the edges of f reversed. We will show that there must exist a forward edge (x', y') in G_f such that $x' \in S$ and $y' \notin S$. This will imply that S cannot be a minimum (s, t)-cut.

Consider a path from u to v in G_f. Since $u \in S$ and $v \notin S$, there must exist an edge (x', y') on this path such that $x' \in S$ and $y' \notin S$. If the edge (x', y') is a backward edge then its reverse (y', x') is a flow edge and, by Observation 11, if S is a minimum (s, t)-cut such that $x' \in S$, then y' must also be in S. Therefore, (x', y') is a forward edge.

Now let us look at the value of the cut S. By the same argument as in the proof of Observation 11, we get

$$\begin{aligned}\text{value}(S) &= \sum_{\substack{(x,y) \in E: \\ x \in S, y \notin S}} c(x,y) \geq \sum_{\substack{(x,y) \in E: \\ \exists i: \ x = x_i, y = y_i \\ \text{or } x = x', y = y'}} c(x,y) \\ &\geq \sum_{i=1}^{d} \phi_i + \phi' = \text{value}(f) + \phi',\end{aligned}$$

where $\phi' > 0$ is the residual capacity of the (forward) edge (x', y') in G_f and x_i, y_i, and ϕ_i are as defined in the proof of Observation 11. Notice that the edge (x', y') is distinct from every (x_i, y_i) since it is a forward edge. Therefore, the value of the cut S is strictly bigger than the value of the maximum flow f, a contradiction with the assumption that S is a minimum cut. □

Therefore, we define a contraction graph that contracts every strongly connected component. We obtain the following corollary.

Definition 13. Let $G = (V, E)$ be a directed graph. We define the *SCC-contraction graph of* G, denoted $\widehat{G}$, to be the graph obtained from G by contracting each strongly connected component into a single vertex.

Corollary 14. *Let* $\widehat{G_f}$ *be the SCC-contraction graph of the residual graph corresponding to an acyclic maximum s-t flow* f *in a flow network* $G = (V, E, c)$. *Suppose* $\hat{s} \in V(\widehat{G_f})$ *is the vertex obtained by contracting the* *strongly connected component containing s and $\hat{t} \in V(\widehat{G_f})$ is the vertex obtained by contracting the strongly connected component containing t. Moreover, we define a function α from the set of $(\hat{t}, \hat{s})$-cuts of $\widehat{G_f}$ to the set of (s,t)-cuts of G: for a $(\hat{t}, \hat{s})$-cut $\hat{T}$ in $\widehat{G_f}$, let $\alpha(\hat{T})$ contain all vertices v that belong to a strongly connected component $\hat{v} \notin \hat{T}$. Then,*

(i) the function α is injective, and

(ii) for every minimum (s,t)-cut S in G there exists a $(\hat{t}, \hat{s})$-cut $\hat{T}$ in $\widehat{G_f}$ such that $\alpha(\hat{T}) = S$.

Proof. To prove (i), let us first look at the (s,t)-cuts S for which there exists a $(\hat{t}, \hat{s})$-cut $\hat{T}$ such that $S = \alpha(\hat{T})$. By the definition of α it follows that any such S must satisfy the property that for every strongly connected component C of G_f, either $C \subseteq S$ or $C \cap S = \emptyset$.

Suppose that we have an S satisfying this property. Then we can uniquely construct $\hat{T}$ such that $S = \alpha(\hat{T})$: for every strongly connected component C of G_f, if $C \cap S = \emptyset$ then the vertex corresponding to C in $\widehat{G_f}$ is in $\hat{T}$ (and otherwise this vertex is not in $\hat{T}$). Since we can reconstruct $\hat{T}$ uniquely, α is injective.

To show part (ii), by Observation 12, every minimum (s,t)-cut S satisfies the property that for every strongly connected component C, either $C \subseteq S$, or $C \cap S = \emptyset$. Therefore, there exists $\hat{T}$ (containing all strongly connected components not in S) such that $S = \alpha(\hat{T})$. □

The next two lemmas show that there is a bijection between minimum (s,t)-cuts in G and forward $(\hat{t}, \hat{s})$-cuts in the SCC-contraction graph. The bijection is given by the function α defined in the above corollary.

Lemma 15. *Under the assumptions of Corollary 14, suppose S is a minimum (s,t)-cut in G. Then $\alpha^{-1}(S)$ is a forward $(\hat{t}, \hat{s})$-cut in $\widehat{G_f}$.*

Proof. Notice that by Corollary 14, $\hat{T} := \alpha^{-1}(S)$ exists (and is unique). By contradiction, suppose that $\hat{T}$ is not a forward $(\hat{t}, \hat{s})$-cut in $\widehat{G_f}$. Then there exists an edge $(\hat{u}, \hat{v}) \in E(\widehat{G_f})$ such that $\hat{v} \in \hat{T}$ and $\hat{u} \notin \hat{T}$.

Since the edge $(\hat{u}, \hat{v})$ is present in the SCC-contraction graph, there must be two vertices u, v in the residual graph G_f such that u is in the strongly connected component corresponding to $\hat{u}$, v is in the strongly connected component corresponding to $\hat{v}$, and there is an edge from u to v in the

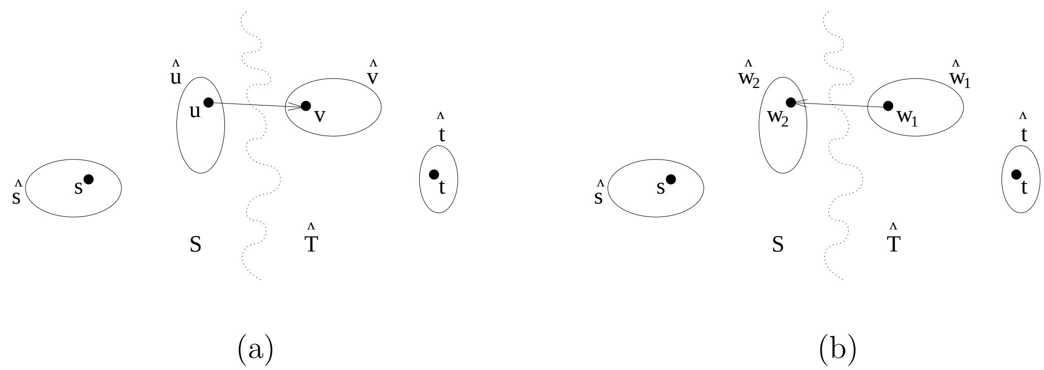

Figure 9: Proofs of Lemmas 15 and 16. The ellipses depict strongly connected components of the residual graph corresponding to f (i. e., the ellipses are vertices of $\widehat{G_f}$), the solid black circles depict vertices of G.

residual graph G_f, see Figure 9(a). This edge is either a forward edge or a backward edge – we will show that either case is impossible.

By the definition of α, since $\hat{v} \in \hat{T}$ and $\hat{u} \notin \hat{T}$, we have that $v \notin S$ and $u \in S$. Suppose (u, v) is a backward edge in G_f. Then, (v, u) is an edge of the flow f. By Observation 11, if S is a minimum (s, t)-cut of G and $u \in S$, then v must be in S as well. We got a contradiction with $v \notin S$.

Alternatively, suppose that (u, v) is a forward edge in G_f. Now let us look at the value of the cut S. By the same argument as in the proof of Observation 11, and since (u, v) is a forward edge and thus differs from all the (x_i, y_i) edges defined in the proof of Observation 11, we get

$$\begin{aligned}\text{value}(S) &= \sum_{\substack{(x,y) \in E:\\ x \in S, y \notin S}} c(x,y) \geq \sum_{\substack{(x,y) \in E:\\ \exists i:\ x = x_i, y = y_i\\ \text{or } x = u, y = v}} c(x,y)\\ &\geq \sum_{i=1}^{d} \phi_i + \phi' = \text{value}(f) + \phi',\end{aligned}$$

where $\phi' > 0$ is the residual capacity of the edge (u, v) in G_f, and x_i, y_i, and ϕ_i are as defined in the proof of Observation 11. Therefore, the value of the cut S is strictly bigger than the value of the maximum flow f, a contradiction with the assumption that S is a minimum cut.

Therefore, the cut $\hat{T}$ must be a forward $(\hat{t}, \hat{s})$-cut in $\widehat{G_f}$. $\square$

Lemma 16. *Under the assumptions of Corollary 14, let $\hat{T}$ be any forward-$(\hat{t}, \hat{s})$-cut in $\widehat{G_f}$. Then the (s, t)-cut $S = \alpha(\hat{T})$ is a minimum (s, t)-cut in G.*

Proof. Recall that by Observation 10, we can decompose the flow f into $d = O(|E|)$ augmenting paths $p_1, \ldots, p_d$. Let us examine the cut value of S. We claim that every edge (u, v) in G such that $u \in S$ and $v \notin S$, is a fully saturated flow edge, and, moreover, for every path p_i there is exactly one edge (u_i, v_i) such that $u_i \in S$ and $v_i \notin S$. From this it follows that the cut value of S is exactly equal to the flow value of f and, therefore, S is a minimum (s, t)-cut of G.

We will first show that every edge (u, v), with $u \in S$ and $v \notin S$, is a fully saturated flow edge, i. e., $c(u, v) = f(u, v)$. By contradiction, suppose that $c(u, v) > f(u, v)$. If $f(u, v) > 0$, then the residual graph G_f contains both the forward edge (u, v) (of residual capacity $c(u, v) - f(u, v) > 0$) and the backward edge (v, u) (of residual capacity $f(u, v)$). Therefore, u and v must belong to the same strongly connected component of G_f. By the definition of α, the vertices u and v must either be both in S, or both not in S – a contradiction.

It remains to deal with the case when $f(u, v) = 0$. Since $c(u, v) > 0$, the edge (u, v) is present as a forward edge in the residual graph G_f. Moreover, we can assume that u, v are not in the same strongly connected component of G_f (if they were, we would achieve a contradiction with $u \in S$ and $v \notin S$). Let $\hat{u}$ be the strongly connected component containing u and let $\hat{v}$ be the strongly connected component containing v. By the definition of α, since $u \in S$ and $v \notin S$ we have that $\hat{u} \notin \hat{T}$, $\hat{v} \in \hat{T}$, and there is an edge from $\hat{u}$ to $\hat{v}$ in $\widehat{G_f}$, see Figure 9(a). This is a contradiction with $\hat{T}$ being a forward $(\hat{t}, \hat{s})$-cut.

Finally, we show that for every path p_i there exists at most one edge (u_i, v_i) such that $u_i \in S$ and $v_i \notin S$. By contradiction, suppose that there would be a path p_i such that there are two edges (u_i, v_i) and (u'_i, v'_i) on p_i such that $u_i, u'_i \in S$ and $v_i, v'_i \notin S$. Suppose that the edge (u_i, v_i) occurs earlier on the path p_i than (u'_i, v'_i) (i. e., the vertex u_i is the closest to s on the path p_i). Then, there must exist an edge (w_1, w_2) on p_i such that $w_1 \notin S$ and $w_2 \in S$. By the definition of α, w_1 and w_2 cannot belong to the same strongly connected component of G_f – let $\hat{w}_1$ be the strongly connected component containing w_1 and let $\hat{w}_2$ be the strongly connected component containing w_2, see Figure 9(b). By the definition of α, we know that $\hat{w}_1 \in \hat{T}$

and $\hat{w}_2 \notin \hat{T}$. Moreover, since (w_1, w_2) is a flow edge, (w_2, w_1) is a backward edge in G_f and therefore $(\hat{w}_2, \hat{w}_1)$ is an edge in the SCC-contraction graph $\widehat{G_f}$. However, since $\hat{w}_1 \in \hat{T}$ and $\hat{w}_2 \notin \hat{T}$, we have a contradiction with $\hat{T}$ being a forward $(\hat{t}, \hat{s})$-cut. □

Finally, we are ready to prove the main reduction theorem.

Proof of Theorem 4. Let f be an acyclic maximum s-t flow in G and let $\tilde{G} = \widehat{G_f}$ be the SCC-contraction graph obtained from the residual graph G_f. Then, the graph $\tilde{G}$ is a directed acyclic graph with $|\tilde{V}| \leq |V|$. Moreover, every edges gives rise to at most one forward edge and at most one backward edge and if both a forward and a backward edge are present in G_f, then the edges lie in the same strongly connected component and neither will appear in $\tilde{G}$. Therefore, $|\tilde{E}| \leq |E|$.

Let $\tilde{s} := \hat{s}$ and $\tilde{t} := \hat{t}$ where $\hat{s}$ and $\hat{t}$ are the vertices corresponding to the strongly connected components containing s and t, respectively. By Lemmas 15 and 16, the forward $(\hat{t}, \hat{s})$-cuts in $\widehat{G_f}$ are in one-to-one correspondence with minimum (s, t)-cuts of G. Moreover, the construction of $\widehat{G_f}$ takes polynomial time: by Observation 8 we spend $O(T(G)+|E|^2)$ time to find f, then we take $O(|E|)$ steps to construct the residual graph G_f and another $O(|E|)$ steps to find its strongly connected components. Therefore, we can construct $\widehat{G_f}$ in time $O(T(G) + |E|^2) = O(|V|^3 + |E|^2)$, see Remark 9. If G is planar, the SCC-contraction graph is also planar since contracting an edge in a planar graph keeps the resulting graph planar. In this case, by Remark 9, the SCC-contraction graph can be constructed in time $O(|V| \log |V|)$, implying the last claim of the theorem. □

Proof of Corollary 5. Since the residual graph is constructed by reversing the flow edges from s to t, for every vertex u that reaches t in G there is a path from u to at least one of the reversed flow edges in G_f. Therefore, u reaches s in G_f. Analogously, every vertex is reachable from t in the residual graph. Therefore, in the contracted graph $\widehat{G_f}$, every vertex reaches $\hat{s}$ and is reachable from $\hat{t}$. This property, along with the graph being acyclic, implies that the only vertex of indegree 0 is $\hat{t}$ and the only vertex of outdegree 0 is $\hat{s}$. □

6. Sampling Minimum (s, t)-Cuts in Planar Graphs

In this section we briefly sketch how to use the counting algorithm to sample minimum (s, t)-cuts in planar graphs uniformly at random. In other words, if Ω is the set of all minimum (s, t)-cuts for a given planar graph, we want to sample each minimum (s, t)-cut with probability $1/|\Omega|$. As we mentioned in the introduction, sampling is related to counting by the well-known reduction due to Jerrum, Valiant, and Vazirani [14]. However, for dynamic-programming-based counting algorithms it is often possible to bypass the reduction (thus avoiding extra factors in the running time).

In our case, the counting algorithm counts all directed paths between a set of pairs of endpoints in a directed acyclic (multi)graph G'_d, see Algorithm 1 and Observation 6. Let $\{(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)\}$ be the set of endpoints and let c_i be the number of paths from a_i to b_i. Then, we generate i proportionally to c_i, to select the pair of endpoints we will use for our random path. Next, we build a random path from a_i to b_i as follows: let $x_1 = b_i$, then for every $j = 1, 2, \ldots$ we select x_{j+1} from the set of immediate predecessors of x_j, proportionally to the number of paths between a_i and the predecessor. We continue this process until we select a_i, then the reverse of $x_1, x_2, \ldots$ forms a uniformly random path from a_i to b_i. (Recall that the algorithm underlying Observation 6 computes all the required information.) If dealing with a multi-graph, we select the predecessor proportionally to the sum of the weights of all paths between a_i and the predecessor.

The sampling procedure takes linear time. Overall, accounting for the time needed to compute the $c_i's$, we need $O(dn + nlogn)$ time to uniformly sample a minimum (s, t)-cut in a weighted planar graph.

Acknowledgments.

We would like to thank the anonymous referees for very useful and detailed feedback.

References

[1] M. O. Ball and J. S. Provan, *Calculating Bounds on Reachability and Connectnedness in Stochastic Networks*, Networks, Vol. 13, 253-278, 1983.

[2] M. O. Ball and J. S. Provan, *Computing Network Reliability in Time Polynomial in the Number of Cuts*, Operations Research, Vol. 32, No. 3, 516-526, 1984.

[3] G. Borradaile and P. Klein, *An $O(n \log n)$ algorithm for maximum st-flow in a directed planar graph*, Journal of the ACM, Vol. 56, Issue 2, 2009.

[4] Y. Boykov and G. Funka-Lea, *Graph Cuts and Efficient N-D Image Segmentation*, International Journal of Computer Vision 70(2), 109-131, 2006.

[5] Y. Boykov and M.-P. Jolly, *Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images*, Proceedings of the Eighth International Conference On Computer Vision (ICCV '01), 105-112, 2001.

[6] Y. Boykov and O. Veksler, *Graph Cuts in Vision and Graphics: Theories and Applications*, in Handbook of Mathematical Models in Computer Vision, edited by Nikos Paragios, Yunmei Chen and Olivier Faugeras, Springer, 2006.

[7] C. J. Colbourn, *Combinatorial aspects of network reliability*, Annals of Operations Research, Vol. 33, Num. 1, 1-15, 2005.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 2001.

[9] L. R. Ford and D. R. Fulkerson, *Maximal flow through a network*, Canadian Journal of Mathematics 8, 399-404, 1956.

[10] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, Journal of the ACM, Vol. 35, Issue 4, 921-940, 1988.

[11] J. Hopcroft and R. E. Tarjan, *Efficient planarity testing*, Journal of the ACM, Vol. 21, Issue 4, 549-568, 1974.

[12] A. Itai and Y. Shiloach, *Maximum Flow in Planar Networks*, SIAM Journal on Computing 8(2), 135-150, 1979.

[13] L. Janiga and V. Koubek, *Minimum Cut in Directed Planar Networks*, Kybernetika, Vol. 28, Num. 1, 37-49, 1992.

[14] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, *Random generation of combinatorial structures from a uniform distribution*, Theoretical Computer Science, Vol. 43, Num. 2-3, 169-188, 1986.

[15] D. R. Karger, *A Randomized Fully Polynomial Time Approximation Scheme for the All-Terminal Network Reliability Problem*, SIAM Journal on Computing 29(2), 492-514, 1999.

[16] J. Kleinberg and É. Tardos, *Algorithm Design*, Addison Wesley, 2005.

[17] H. Nagamochi, Z. Sun, and T. Ibaraki, *Counting the number of minimum cuts in undirected multigraphs*, IEEE Transactions on Reliability, Vol. 40, Issue 5, 610-614, 1991.

[18] J. S. Provan and M. O. Ball, *The complexity of counting cuts and of computing probability that a graph is connected*, SIAM Journal on Computing 12(4), 777-788, 1983.

[19] A. Ramanathan and C. J. Colbourn, *Counting almost minimum cutsets with reliability applications*, Mathematical Programming, Vol. 39, Num. 3, 253-261, 1987.

[20] J. H. Reif, *Minimum s-t cut of a planar undirected network in $O(n \log^2 n)$ time*, SIAM Journal on Computing 12, 71-81, 1983.