Maximum *st*-flow in directed planar graphs via shortest paths

Glencora Borradaile and Anna Harutyunyan
Oregon State University

Abstract. Minimum cuts have been closely related to shortest paths in planar graphs via planar duality – so long as the graphs are undirected. Even maximum flows are closely related to shortest paths for the same reason – so long as the source and the sink are on a common face. In this paper, we give a correspondence between maximum flows and shortest paths via duality in directed planar graphs with no constraints on the source and sink. We believe this a promising avenue for developing algorithms that are more practical than the current asymptotically best algorithms for maximum *st*-flow.

Keywords: maximum flows, shortest paths, planar graphs

1 Introduction

The asymptotically best algorithm for max *st*-flow in directed planar graphs is due to Borradaile and Klein [1]. The algorithm is the *leftmost augmenting-path algorithm*; in each iteration, flow is pushed along the leftmost path from *s* to *t*. There are at most 2*n* iterations [4] and, using a dynamic-trees data structure, each iteration can be implemented in \( O(\log n) \) time (e.g. [18]). The algorithm is, in fact, a generalization of the algorithm first suggested by Ford and Fulkerson in their seminal *Max-Flow, Min-Cut* paper for the planar case when *s* and *t* are on a common face, or *st*-planar [5]. However, in the *st*-planar case, each iteration of the leftmost-augmenting path algorithm can be implemented in \( O(\log n) \) time using priority queues [6].

Priority queues are arguably simpler and more practical than dynamic trees. In fact, Tarjan and Werneck have shown experimentally that a naïve, linear-time-per-operation implementation of dynamic-trees outperforms more sophisticated logarithmic-time implementations in many scenarios [20]. The reason that priority queues are sufficient for the *st*-planar case is because the leftmost-augmenting paths algorithm, in this case, can be implemented as Dijkstra’s algorithm in the dual graph. We propose and pursue the following:

Conjecture 1. An augmenting-path algorithm for max *st*-flow in directed planar graphs can be implemented with \( O(n) \) queries to a priority-queue.

In this paper, we make progress toward this conjecture by showing that the leftmost augmenting-path algorithm in the general case can be reduced to the *st*-planar case in a covering graph.

\(^1\) Formal definitions are delayed until the next section.
1.1 Related results

The relationship between shortest paths, minimum cuts and maximum flows in planar graphs has been studied from very early in the history of maximum flow algorithms. Whitney showed that a minimum st-cut corresponds to a shortest st-separating cycle in the graph’s dual [21]. Hassin showed that for st-planar graphs, by splitting the face common to s and t in two (becoming two vertices in the dual), finding a shortest-path tree in the dual (rooted at one of the new vertices\(^2\)) provides all the information needed for the maximum st-flow [6].

When s and t are not on the same face, the correspondence between shortest paths and maximum flows is weaker. In fact, the known relationships are indirect, reducing max flow or min cut to a sequence of $O(n)$ or $O(\phi)$ st-planar max flow or min cut computations. Reif showed that when the graph is undirected, there is a set of at most $n$ nesting, separating cycles in the dual, one of which is the minimum st-separating cycle [17]; these cycles can be found via divide and conquer, each with a shortest-path computation. Hassin and Johnson showed that the shortest-path distances computed in Reif’s algorithm are sufficient to reconstruct a maximum flow [7]. Kaplan and Nussbaum reduced the number of cycles required for Reif’s algorithm to at most $\phi$, where $\phi$ is the minimum number of faces that any s-to-t curve must pass through [14]. Unfortunately, the $\phi$ shortest-path tree distances computed are insufficient for reconstructing the flow. The only result for maximum st-flows or minimum st-cuts in directed planar graphs that utilizes shortest-path computations in the dual, to our knowledge, is an approach used by Itai and Shiloach [11] and Johnson and Venkatesan [13] which infeasibly routes flow along an artificial path of length $\phi$ and reduces the problem to $\phi$, sequentially-computed st-planar maximum flows. (A summary of these results is provided in Table 1 in Appendix A.)

1.2 Overview

In Section 2, we define a covering graph by cutting open a cylindrical embedding of the graph along an s-to-t curve and pasting copies of the resulting plane together. This covering graph is similar to that used by Erickson [4] for presenting a simpler proof of the bound for Borradaile and Klein’s algorithm. In Section 3, we show that the leftmost flow in the covering graph that contains $O(\phi)$ copies of the original graph contains the leftmost flow in the original graph.\(^3\)

In terms of running time, our algorithm does not improve on that of Johnson and Venkatesan’s result mentioned above [13]. Further, the space requirement for implementing our algorithm would be an $O(\phi)$ factor higher. However, the algorithmic technique behind our algorithm is amenable to the implementation of an augmenting-path algorithm using only priority queues (and not dynamic trees). We discuss this more in Section 4 and provide more technical evidence for this avenue of research in Appendices E and F. The algorithms in these

\(^2\) For details of Hassin’s algorithm, see Appendix D.

\(^3\) We need to first find the value of the flow, but finding the value of the flow uses the same techniques.
appendices do not suffer from the additional space requirement. In fact, the algorithm in Appendix F gives an implementation of Borradaile and Klein’s algorithm which results in at most \((2\phi + 2)n\) priority queue queries (or at most \(2\phi + 2\) shortest-path computations). Unlike other algorithms depending on the parameter \(\phi\), this algorithm does not need to know the value of \(\phi\), and may use many fewer iterations.

We cannot fail to mention that shortest paths (and so, \(st\)-planar flows) can be computed in \(O(n)\) time [8]. However, this linear-time algorithm is very complicated and, to our knowledge, has never been implemented – it is very likely that Dijkstra’s algorithm would prove to be more practical in most situations. Further, Dijkstra’s algorithm is more amenable to modifications; in a companion paper, we use a modification of Dijkstra’s algorithm for planar graphs in a situation where the linear-time algorithm cannot be used.

1.3 Background

The graphs of this paper are directed, but we consider the underlying undirected graph. Each edge of the undirected graph is composed of oppositely-directed darts each oriented from \(\text{tail} \rightarrow \text{head}\); \(\text{rev}(\cdot)\) maps a dart to its reverse. Capacities \(c\) are non-negative and defined over the set of darts. Paths and cycles are sequences of darts, and so are naturally directed. Paths may visit the same vertex multiple times; those that do not are simple. A path may be trivial in which case it is a single vertex. We use \(X[a,b]\) to denote subpath of path or cycle \(X\) that include the endpoints \(a\) and \(b\). We use \(\circ\) to denote the concatenation of paths.

We extend any function or property on elements to sets of elements in the natural way.

**Flows** A flow assignment \(f\) is an assignment of real numbers to the darts of \(G\) that is antisymmetric, i.e., \(f[d] = -f[\text{rev}(d)]\). A pseudoflow is a flow assignment that respects the capacity of every dart \(d\): \(f[d] \leq c[d]\) [9]. The net flow of a vertex \(v\) is the sum of the flow of darts whose head is \(v\). Vertices with positive net flow are excess vertices; vertices with negative net flow are deficit vertices. A pseudoflow with zero net flow at every vertex is a circulation. An \(ST\)-flow or simply, a flow, is a pseudoflow whose only deficit vertices are in \(S\), the sources, and only excess vertices are in \(T\), the sinks. We consider mostly the single-source, single-sink case in which \(S = \{s\}\) and \(T = \{t\}\) and which we refer to as an \(st\)-flow. The value of a flow is the sum net flows of its sources and is denoted \(|f|\).

Given capacities \(c\) and a flow assignment \(f\), the residual capacity of dart \(d\) is \(c_f[d] = c[d] - f[d]\). A dart \(d\) is residual if \(c_f[d] > 0\). We say that a set of darts is saturated if one dart in the set is non-residual (giving saturated path, cycle, residual flow).

\footnote{As the original graph is directed, \(c[d]\) need not be equal to \(c[\text{rev}(d)]\). For a directed edge \(a\) of the original graph with capacity \(c\), \(c[d] = c\) and \(c[\text{rev}(d)] = 0\) where \(d\) is the dart in the underlying undirected graph in the direction of \(a\).}
etc.). Residuality is with respect to a set of capacities; those capacities may be
the original capacities or the residual capacities as indicated in the text.

A flow is maximum if and only if there are no residual source-to-sink
paths \([5]\).

A pseudoflow is maximum if and only if there are no residual paths from
a source or excess vertex to a deficit or sink vertex \([9]\).

We say that a set of darts are flow darts if each dart in the set has positive
flow (giving flow path, cycle, etc.). We say that a flow assignment is acyclic if
there are no flow cycles. Reducing the flow on a set of flow darts is the operation
of reducing the flow assignment on those darts by a fixed amount (and changing
the flow assignment on the reverse of those darts accordingly). Acyclic, maxi-
mum pseudoflows can be converted into maximum flows by reducing the flow on
source-to-excess and deficit-to-sink flow paths until all but the sources and sinks
are balanced. Although we do not use this fact algorithmically, this can be done
in linear time \([2, 12]\).

Planar graphs We use the usual definitions for a planar embedded graph \(G\)
and its dual \(G^*\). See Ref. \([3]\), e.g., for formal definitions.

Suppose \(a, b, \) and \(d\) are darts such that \(\text{head}(a) = \text{tail}(b) = \text{head}(d) = v\); we
say \(d\) enters \(a \circ b\) at \(\text{head}(a)\). If the clockwise ordering of these darts around \(v\)
in the embedding is \(a, b, d\), then \(d\) enters \(a \circ b\) from the right and \(\text{rev}(d)\) leaves
\(a \circ b\) from the right. Entering and leaving from the left are defined analogously.
We say that \(P\) crosses \(Q\) from right to left at \(X\) if \(X\) is a maximal subpath \(5\) of
\(Q\) such that either \(X\) or \(\text{rev}(X)\) is a subpath of \(P\) and \(P\) enters \(Q\) from the right
at \(\text{start}(X)\) and \(P\) leaves \(Q\) from the left at \(\text{end}(X)\). Crossing from left to right
is defined analogously. By definition, \(X\) cannot be a prefix or suffix of either \(P\)
or \(Q\). If \(P\) and \(Q\) are paths that do not cross, then they are non-crossing. A
path/cycle is non-self-crossing if for every pair \(P\) and \(Q\) of its subpaths, \(P\) does
not cross \(Q\). The notions of entering and crossing are illustrated in Appendix \([8]\).

We define a set of darts left \((P)\) that are on the left side of a path \(P\). left \((P)\)
is a subset of the darts whose head or tail (but not both) \(6\) is in \(P\). For a dart \(d\)
whose head or tail is in \(P\) but is not an endpoint of \(P\), \(d \in \text{left}(P)\) if \(d\) enters
or leaves \(P\) from the left. If \(d\)'s head or tail is an endpoint of \(P\), we break the
left/right tie arbitrarily but consistently as follows. For each vertex \(x\), we assign
an arbitrarily chosen face \(f_x\) adjacent to \(x\). Suppose \(P\) is an \(s\)-to-\(t\) path. If the
head or tail of \(d\) is \(s\) (resp. \(t\)), then \(d \in \text{left}(P)\) if \(d\) is in the c.w. (resp. c.c.w. )
ordering of the darts and faces around \(s\) (resp. \(t\)) in the embedding between \(f_s\)
(resp. \(f_t\)) and the first (resp. last) dart of \(P\). right \((P)\) is defined analogously.

We define the graph \(G \mathcal{X} P\) as the graph cut open along \(P\). \(G \mathcal{X} P\) contains
two copies of \(P\), \(P_L\) and \(P_R\), so that that the edges in left \((P)\) are adjacent to
\(P_R\) and the edges in right \((P)\) are adjacent to \(P_L\).

---

\(5\) For a trivial path \(X\) that is a single vertex \(x\), start\((X)\), end\((X)\) and \(\text{rev}(X)\) are \(x\)
itself.

\(6\) Assuming, w.l.o.g. that there are no parallel edges.
Finally, we remind the reader of the parameter $\phi$ defined in the introduction and which we will use throughout: $\phi$ is the minimum number of faces that any curve drawn in the surface in which $G$ is embedded that any $s$-$t$ path must go through.

**Clockwise and leftmost** A potential assignment $\phi$ is an assignment of real numbers to the faces of a planar graph. Corresponding to every circulation in a planar graph, there is a potential assignment such that the flow on a dart $d$ is given by the difference between the face on the right side and left side of $d$. We take the potential of the infinite face to be 0. A circulation is clockwise (c.w.) if all the potentials are non-negative. A cycle $C$ is clockwise if the circulation that assigns +1 to every dart in $C$ and -1 to the reverse of these darts (and 0 otherwise) is clockwise. The definition of clockwise depends on the choice of $f_\infty$.

Throughout this paper, we will take $f_\infty = f_t$.

An $s$-$t$ path $A$ is left of an $s$-$t$ path $B$ if $A \cup \text{rev}(B)$ is a clockwise circulation. A path is leftmost if there are no paths left of it. A simple leftmost path from $s$-$t$ can be found by a depth-first left-most search [1]. A flow assignment is leftmost if every clockwise cycle is non-residual. Leftmost paths and circulations are unique [16]. The leftmost $st$-flow of a given value is unique for the same reason. The next lemma follows immediately from the definitions:

**Lemma 1.** A leftmost circulation can be decomposed into a set of flow-carrying clockwise simple cycles.

Khuller, Naor and Klein illustrated that the leftmost circulation corresponds to the potential assignment given by shortest-path distances in the dual (interpreting capacities as lengths) [16]. It follows immediately that the maximum flow given by Hassin’s algorithm, from dual shortest-path distances in the dual, is leftmost.

We say that (residual) capacities $c$ are c.w. acyclic if there are no c.w. cycles that are residual. In our algorithms, we start with the residual capacities obtained via dual shortest paths according to Khuller, Naor and Klein; these are c.w. acyclic. Leftmost flow assignments w.r.t. c.w. acyclic capacities are acyclic [1].

**Theorem 1.** Let $L$ be the leftmost residual $s$-$t$-path in $G$ w.r.t. c.w. acyclic capacities $c$. Let $f$ be any $st$-flow (of any value). Then no simple $s$-$t$ flow path crosses $L$ from the left to right.

**Proof.** Let $F$ be an $st$-flow path. Suppose for a contradiction that $F$ crosses $L$ from left to right at $X$ (a subpath of $L$). Since $F$ and $L$ both start at $s$, $L[s, \text{start}(X)]$ must contain a vertex of $F$. Let $v$ be the last such vertex. Since $F$ is a flow path, $F$ must be residual w.r.t. $c$ (in order for flow to be routed on it). Since $L$ is the leftmost residual path w.r.t. $c$, no subpath of $F$ can be left of any subpath of $L$ (between the same endpoints). It follows that $L[v, \text{start}(X)] \circ \text{rev}(F[v, \text{start}(X)])$ is a simple clockwise cycle. Further, since

\[^{7}\text{Not necessarily leftmost.}\]
$F$ crosses $L$ from left-to-right at $X$, $F$ enters the strict interior of this cycle. (See figure in Appendix B.) To escape (to get to $t$), $F$ must hit a vertex $u$ of $L[u, \text{start}(X)]$ (since $F$ is simple). Then $F[\text{start}(X), u] \circ L[u, \text{start}(X)]$ is a clockwise cycle, contradicting that the initial capacities $c$ are c.w. acyclic.

2 Infinite covers

In our analysis, we use an infinite covering graph of $G$ derived from the universal cover of a cylinder on which we embed $G$. In our algorithm, we use a finite subgraph of this infinite covering graph. The construction of our cover is similar to the one described in Section 2.4 of Erickson’s analysis of the leftmost-path algorithm of Borradaile and Klein. Erickson’s cover was of the dual graph, whereas we remain in the primal as our analysis remains entirely in the primal.

Embed $G$ on a sphere and remove the interiors of $f_t$ and $f_s$. The resulting surface is a cylinder with $t$ and $s$ embedded on opposite ends. The universal cover of this cylinder is an infinite strip. The repeated drawing of $G$ on the universal cover of the cylinder defines a covering graph $\tilde{G}$ of $G$. For a subgraph $X$ of $G$, we denote the subgraph of $\tilde{G}$ whose vertices and darts map to $X$ by $\tilde{G}[X]$. We say $\tilde{X} \subset \tilde{G}[X]$ is a copy of $X$ if $\tilde{X}$ maps bijectively to $X$. We say that $\tilde{X}$ is an isomorphic copy of $X$ if $\tilde{X}$ is isomorphic to $X$. Note that an isomorphic copy need not exist. (For example, there is no isomorphic copy of $G$ in $\tilde{G}$, if $G$ is not $st$-planar.)

For ease of notation, we take the top (resp. bottom) of the cylinder to correspond to $f_t$ (resp. $f_s$); all the vertices in $\tilde{G}[t]$ (resp. $\tilde{G}[s]$) are on the top (resp. bottom) of the infinite strip. We can therefore refer to the left and right ends of the strip (which extend to infinity). We number the copies of $t$ from left to right: $\tilde{G}[t] = \{ \ldots t^{-1}, t^0, t^1, \ldots \}$, picking $t^0$ arbitrarily. For a simple $s$-to-$t$ path $P$ in $G$, we denote by $P^i$ the isomorphic copy of $P$ in $\tilde{G}[P]$ that ends at $t^i$. $P^i$ divides the infinite strip into two sides which are the components of $\tilde{G} \setminus P^i$, the portion containing the left end and the portion containing the right end. Therefore, we can order the copies from left to right, i.e., $\tilde{G}[P] = \{ \ldots , P^{i−1}, P^0, P^1, \ldots \}$ where $P^i$ is in the left component of $\tilde{G} \setminus P^j$ for all $i < j$.

We further define copies of $G$ w.r.t. $P$: $G_P^i \cup P^{i+1}$ is the finite component of subgraph of $\tilde{G} \setminus P^j \setminus P^{j+1}$.

Observation 1 $G_P^i \cup P^{i+1}$ is isomorphic to $G \setminus P$ with $P^i$ mapped to $P_L$ and $P^{i+1}$ mapped to $P_R$.

We also number the copies of $s$ from left to right: $\tilde{G}[s] = \{ \ldots s^{-1}, s^0, s^1, \ldots \}$, picking $s^0$ arbitrarily. (In the next section, for convenience of notation, we will make this choice linked to the choice of $t^0$.)

The following lemma connects the relative topology of two paths in $G$ to that in $\tilde{G}$.

Lemma 2. Let $P$ be a simple $s$-to-$t$ path and let $Q$ be a simple path sharing only its endpoints with $P$. Consider an isomorphic copy $\tilde{Q}$ of $Q$ in $\tilde{G}$. Let $a$ and $b$ be the first and last darts of $Q$, respectively.
1. If \( a, b \in \text{left}(P) \), \( Q \) starts and ends on \( P^j \) for some \( j \).
2. If \( a, b \in \text{right}(P) \), \( Q \) starts and ends on \( P^j \) for some \( j \).
3. If \( a \in \text{left}(P) \), \( b \in \text{right}(P) \), \( Q \) starts on \( P^j \) and ends on \( P^{j+1} \) for some \( j \).
4. If \( a \in \text{right}(P) \), \( b \in \text{left}(P) \), \( Q \) starts on \( P^j \) and ends on \( P^{j+1} \) for some \( j \).

Proof. Consider \( Q \) in \( G \times P \). Since \( Q \) shares only its endpoints with \( P \), \( Q \) is path in \( G \times P \). For the 4 cases of the lemma, by definition of \( \mathcal{L} \), \( Q \) is a (1) \( P_{R}-P_{R} \), (2) \( P_{L}-P_{L} \), (3) \( P_{R}-P_{L} \), and (4) \( P_{L}-P_{R} \) path. The lemma then follows from Observation[1]. \(\square\)

The following two lemmas relate clockwise cycles in \( \mathcal{G} \) and in the original graph \( G \). The following holds for any simple \( s\)-to-\( t \) path in \( P \).

**Lemma 3.** Let \( u^i \) and \( u^j \) be copies in \( \mathcal{G} \) of the vertex \( u \) in \( G \) such that \( u^i \in G^i_P \) and \( u^j \in G^j_P \). Let \( Q \) be any \( u^i\)-to-\( u^j \) path in \( \mathcal{G} \). If \( i < j \), then the mapping of \( Q \) into \( G \) contains a clockwise cycle.

Proof. Suppose \( i < j \) and consider the composition of copies of \( Q \) to create an infinite path \( Q \) in \( \mathcal{G} \). \( Q \) is a \(-\infty\)-to-\( \infty \) path and may not be simple. If we imagine a source at \(-\infty \) and a sink at \( \infty \) and we send one unit of flow along \( Q \), we have a flow of value 1. It follows that there is a simple \(-\infty\)-to-\( \infty \) path \( R \) that contains a subset of the darts of \( \mathcal{G} \).

Consider the potential function \( \Phi_G \) over the faces of \( \mathcal{G} \) that assigns 1 to every face below \( R \) and 0 to all other faces. Define the potential function \( \Phi_G \) over the faces of \( G \) where \( \Phi_G[f] = \max_{f \in \mathcal{G}} f \Phi_G[f] \) for face \( f \) of \( G \). By definition of c.w. circulation, \( \Phi_G \) defines a clockwise circulation (for \( f_{\infty} = f_1 \)). By definition of \( \Phi_G \), only darts mapped to by \( R \) are flow darts. By Lemma[1] there is a c.w. flow cycle in the circulation. \(\square\)

The proof of the following lemma is very similar to that for Lemma[2] and omitted in the efforts of brevity:

**Lemma 4.** Let \( C \) be a simple clockwise cycle in \( \mathcal{G} \). A subset of the darts of \( C \) maps to a clockwise cycle in \( G \).

The following lemma is key in bounding the size of the finite portion of \( \mathcal{G} \) that we will need to consider in our algorithms.

**Lemma 5 (Pigeonhole).** Let \( P \) be a simple path in \( G \). Let \( \bar{P} \) be an isomorphic copy of \( P \) in \( \mathcal{G} \). \( \bar{P} \) contains a dart of at most \( \phi + 2 \) copies of \( G \) in \( \mathcal{G} \). If \( P \) may only use \( s \) and \( t \) as endpoints, then \( \bar{P} \) contains darts in at most \( \phi \) copies of \( G \) in \( \mathcal{G} \).

Proof. Let \( \Pi \) be the smallest set of faces that a curve from \( s \) to \( t \) drawn on the plane in which \( G \) is embedded must cross. Embed in \( G \) an artificial edge in each of these faces to connect \( s \) to \( t \); we refer to this path as \( \Pi \) as well; \( \Pi \) has \( \phi + 1 \) vertices.

Consider an isomorphic copy \( \bar{P} \) of \( P \) in \( \mathcal{G} \). Let \( i (j) \) be the minimum (maximum) index such that \( \bar{P} \) contains a vertex \( u^i (u^j) \) of \( G^i_{\Pi} (G^j_{\Pi}) \). Since \( \bar{P} \) must have
darts in copies $i, \ldots, j$ of $G$. $P$ must contain a vertex of $\Pi^k$ for $k = i + 1, \ldots, j$. If $j - (i + 1) > \phi + 1$, then by the pigeonhole principle, $P$ must visit two copies of the same vertex of $\Pi$, contradicting that $P$ is simple. Therefore, $j - i \leq \phi + 2$, proving the first part of the lemma.

Suppose $P$ may only use $s$ or $t$ as endpoints. Let $P'$ be the subpath of $P$ with $s$ and/or $t$ removed. Now, defining $i$ and $j$ as above, $P'$ visits the two copies of the same vertex if $j - (i + 1) > \phi - 1$ as $P$ does not contain $s$ or $t$. Therefore $j - i \leq \phi$, proving the second part of the lemma. □

3 Maximum flow, shortest paths equivalences

We are ready to illustrate an equivalence between dual shortest paths and maximum flow. We assume that our initial capacities $c$ are c.w. acyclic. We identify a finite portion of $G$ (containing $k$ copies of $G$), $G_k$ (Section 3.1). In this finite cover, we compute the leftmost maximum flow: We first attach a super-source $S$, embedded below the cover, to each source with a large-capacity arc and a super-sink $T$, embedded above the cover, to each sink with a large-capacity arc (creating graph $G_{ST}^k$ and capacities $c_{ST}^k$). We can then find the leftmost max $ST$-flow $f_{ST}^k$ via Hassin’s method. We show how to extract from $f_{ST}^k$ the value of the maximum $st$-flow $|f|$ in $G$ (Section 3.1). Given this value, we are able to change the capacities $c_{ST}^k$ so that we can extract $f$ from $f_{ST}^k$ (Section 3.3).

This method requires a factor $k$ additional space. In Section 4, we discuss how this additional space requirement could be removed and how a dual shortest-path algorithm could be used to simulate an augmenting-paths algorithm in $G$ even though $s$ and $t$ are not on a common face. This, we believe, is a promising avenue for developing a practical algorithm for maximum flow in planar graphs.

Our proof technique is as follows. We can, using dual-shortest path techniques, compute $f_{ST}^k$, the leftmost maximum $ST$-flow in $G_{ST}^k$. We wish to find $f$, the leftmost maximum $st$-flow in $G$. We relate $f_{ST}^k$ to $f$ by first copying $f$ into $G_{ST}^k$ and then modifying the resulting flow, without changing the flow assignment in the central copy of $G$ of $G_{ST}^k$. We can therefore extract $f$ from the flow assignment given by $f_{ST}^k$ in this central copy of $G$.

3.1 The finite cover

Let $L$ be the leftmost residual $s$-to-$t$ path in $G$ and let $f$ be the leftmost maximum $st$-flow in $G$; since the capacities are c.w. acyclic, $f$ is acyclic.

Let $G_k$ be the finite component of $G \times L^0 \times L^k$; $G_k$ is a finite cover made of $k$ copies of $G$ (plus an extra copy of $L$), bounded on the left by $L$ and on the right by the extra copy of $L$. The sinks of $G_k$ are numbered $t^0, t^1, \ldots, t^k$ from left to right; we (re)number the sources from left to right, $s^0, s^1, \ldots, s^k$. We will refer to the $1$st through $k$th copies of $G$ in $G_k$ according to the natural left-to-right ordering.

We start by constructing a maximum multi-source, multi-sink maximum flow in $G_k(f_1)$, from a maximum pseudoflow $f_0$ (Lemma 6), which is constructed from
There are no paths. The proof for the first two cases are symmetric; we only prove one here. Let $f_0$ be a flow assignment for $G_k$ given by $f_0[d] = f[d], \forall d \in G[d]$. We overload $c$ to represent capacities in both $G$ and $G_k$, where capacities in $G_k$ are inherited from $G$ in the natural way.

**Lemma 6.** For $k > \phi + 2$, $f_0$ is a maximum pseudoflow with excess vertices on $L^0$ and deficit vertices on $L^k$.

**Proof.** First notice that $f_0$ is balanced for all vertices in $G_k$ except those on $L^0$ and $L^k$; they may contain excess and deficit vertices. By Lemma 4 there are no flow paths in $f$ that push flow from the left of $L$ to the right of $L$. It follows that $L^0$ only contains vertices with excess or balanced vertices, and $L^k$ only contains vertices with deficit or balanced vertices.

In $G_k$ and $G$, we use residual to mean w.r.t. $c_f$ and $c_f$, respectively. A pseudoflow is maximum if there is no $S \cup V^+\to T \cup V^- \to$ residual path $P$, where $V^+$ is the set of excess vertices and $V^-$ is the set of deficit vertices. If $P$ is a path from a source to a sink in $G_k$, then $P$ maps to an $s$-$t$ path in $G$; therefore, $P$ cannot be residual in either graph.

It remains to show that there are no $V^+\to T, S\to V^-$ or $V^+\to V^-$ residual paths. The proof for the first two cases are symmetric; we only prove one here.

There are no $V^+\to T$ residual paths. Consider for a moment the flow assignment for $G$: $f'[d] = f[d], \forall d \in G[d]$. For $v^+ \in V^+$ to be an excess vertex, there must be a $v\to t$ flow path $Q$ in $f$ where $v$ is the vertex in $G$ that $v^+$ maps to. There is a copy $Q$ of $Q$ in $G$ that starts at $v^+$ (where we remind the reader that $G_k$ is a subgraph of $G$). By Lemma 4 $Q$ does not cross $L$ from left to right and so $Q$ is left of $L^0$.

Now, for a contradiction, let $R$ be a $v^+\to t^+$ residual path, for some $t^+ \in T$. Since the reverse of a flow path is residual, $\text{rev}(Q) \circ R$ is a residual $v^-\to t^+$ path in $G$ (w.r.t. $f'$). Since $Q$ is left of $L^0$, $j \leq i$. If $j = 0$, $Q \circ \text{rev}(L^0[v^+, t^0])$ would be a clockwise cycle, which, by Lemma 3 would witness a clockwise cycle in $G$; since $Q$ is residual w.r.t. the original capacities $c$, this cycle would contradict the leftmost-ness of $L$. Therefore $j < i$. By Lemma 3 $\text{rev}(Q) \circ R$ implies the existence of a clockwise residual cycle in $G$, contradicting the leftmost-ness of $f$.

There are no $V^+\to V^-$ residual paths. Let $v^+$ and $v^-$ be vertices in $V^+$ and $V^-$, respectively. A $v^+\to v^-$ path $R$ must go through all $k$ copies of $G$ in $G_k$. Since $k > \phi + 2$, by the Pigeonhole Lemma, $R$ contains two copies $u^i$ and $u^j$ of the same vertex $u$ in $G$, and in fact, must contain a subpath from $u^i$ to $u^j$ for $i < j$. By Lemma 3 this implies that there is a clockwise residual cycle in $G$, contradicting $f$ being leftmost.

**Lemma 7.** There is a maximum $ST$-flow $f_1$ in $G_k$ that is obtained from $f_0$ by removing flow on darts in the first and last $\phi$ copies of $G$ in $G_k$. Further, the
amount of flow into sink \( t' \) for \( i \leq k - \phi \) and the amount of flow out of source \( s^j \) for \( j \geq \phi \) is the same in \( f_0 \) and \( f_1 \).

Proof. Since \( c \) are clockwise-acyclic capacities, and \( f \) is leftmost, \( f \) is acyclic. It follows that \( f_0 \) is acyclic. Since \( f_0 \) is an acyclic maximum pseudoflow, it can be converted to a maximum flow by flow-cancelling techniques [9]; i.e., by removing flow from source-to-excess flow paths and deficit-to-sink flow paths. Let \( P \) be such a flow path. \( P \) maps to a flow path in \( G \) and so must map to a simple path in \( G \). By the Pigeonhole Lemma, \( P \) must be contained within \( \phi \) copies of \( G \). This proves the first part of the lemma. Since \( P \) cannot start at \( s^j \) for \( j \geq \phi \) without going through more than \( \phi \) copies (and likewise, \( P \) cannot end at \( t' \) for \( i \leq k - \phi \)), the second part of the lemma follows. \( \square \)

3.2 Value of the maximum flow

In the next lemma, we prove that from \( f^{ST} \), the leftmost maximum \( ST \)-flow in \( G_k^{ST} \), we can extract \(|f|\), the value of the maximum \( st \)-flow in \( G \).

Lemma 8. For \( k \geq 4\phi \), the amount of flow through \( s^{2\phi} \) in \( f^{ST} \), the leftmost maximum flow in \( G_k^{ST} \), is \(|f|\).

Proof. We show that the amount of flow leaving \( s^{2\phi} \) in \( f^{ST} \) is the same as in \( f_1 \). By Lemma 7, the amount of flow leaving \( s^{2\phi} \) is the same in \( f_1 \) as \( f_0 \) which is the same as the amount of flow leaving \( s \) in \( f \); this proves the lemma.

First extend \( f_1 \) into an \( ST \)-flow, \( f_1^{ST} \), in \( G_k^{ST} \) in the natural way (by setting the flow on the darts adjacent to \( S \) and \( T \) to satisfy the balance constraint). Since \( f_1 \) is a maximum multi-source, multi-sink flow in \( G_k \), \( f_1^{ST} \) is a max \( ST \)-flow in \( G_k^{ST} \).

To convert \( f_1^{ST} \) into a leftmost flow, we must saturate the clockwise residual cycles; this is done with a c.w. circulation, which, by Lemma 4, can be converted into a set of flow carrying c.w. simple cycles. Suppose for a contradiction that one such cycle \( C \) changes the amount of flow through \( s^{2\phi} \). Since \( C \) is a flow cycle in the circulation (which is residual w.r.t. \( c_{f^{ST}} \)), \( C \) is residual w.r.t. \( c_{f^{ST}} \). If \( C \) changes the amount of flow through \( s^{2\phi} \), \( C \) must go through \( S \). \( C \) cannot visit \( T \), for if it did, \( C \) would include a source-to-sink path. This path is also residual w.r.t. \( c_{f^{ST}} \), contradicting that \( f_1^{ST} \) is maximum. Therefore \( C \) must contain a \( s^i \)-to-\( s^{2\phi} \) path \( P \) that is in \( G_k^c \); \( P \) is residual w.r.t. \( c_{f_1} \). Since \( C \) is c.w., \( i < 2\phi \).

We first argue that \( P \) must use a dart in the first or last \( \phi \) copies of \( G \) in \( G_k^c \). Suppose otherwise. Then \( P \) must map to a set \( P' \) of darts in \( G \) which, by Lemma 5, are residual w.r.t. \( c_{f} \). By Lemma 3 \( P' \) contains a clockwise cycle, contradicting the leftmostness of \( f \).

Therefore, \( P \) must visit a dart in the first or last \( \phi \) copies of \( G \). It follows that \( P \) must cross either from the \( \phi^i \)-th copy to \( s^{2\phi} \) or from \( s^{2\phi} \) to the \( 3\phi^i + 1 \)-th copy (possible, since \( k \geq 4\phi \)). Then, by the Pigeonhole Lemma, \( P \) contains a subpath \( Q \) that goes from \( \bar{v} \) to \( \bar{v'} \) where these vertices are copies of the same vertex that are not in the first or last \( \phi \) copies and such that \( \bar{v} \) is in an earlier
Maximum \textit{st}-flow in directed planar graphs via shortest paths

3.3 Maximum flow

Now, suppose we know $|f|$ (as per Lemma 3.2). We change the capacities of the arcs into $T$ and out of $S$ in $G_k^{ST}$ to $|f|$. Call these capacities $c[|f|]$. Now, $f^{ST}$, as defined in the previous section, respects $c[|f|]$ since, by Lemmas 6 and 7, the amount of flow leaving any source or entering any sink in $f^1$ is at most $|f|$. We prove a lemma similar to Lemma 7. The proof of this lemma is similar to the proof of Lemma 8 so we defer it to Appendix C.

**Lemma 9.** $f^{ST}$ can be converted into a leftmost maximum \textit{ST}-flow $f[|f|]$ for the capacities $c[|f|]$ while not changing the flow on darts in the first or last $2\phi$ copies of $G$ in $G_k$.

To summarize, Lemmas 7 and 9 guarantee that the maximum leftmost \textit{ST}-flow, $f[|f|]$, in $G_k^{ST}$ given capacities $c[|f|]$ has the same flow assignment on the darts in copy $2\phi + 1$ as $f$ so long as $k \geq 4\phi + 1$. Starting from scratch, we can find c.w. acyclic capacities $c$ via Khuller, Naor and Klein’s method [16] (one shortest path computation); we can find $|f|$ (Lemma 8, a second shortest path computation) and then $f$ (Lemma 9, a third shortest path computation). Therefore, finding a maximum \textit{st}-flow in a directed planar graph $G$ is equivalent to three shortest path computations: one in $G$ and two in a covering of $G$ that is $4\phi + 1$ times larger than $G$.

4 Discussion

Borradaile and Klein’s leftmost augmenting-paths algorithm also uses c.w. acyclic capacities as a starting point [1]. Their analysis showed that an arc can be augmented and possibly its reverse, but, when this happens, the arc becomes unusable and will not be a part of any further augmenting path. In Appendix F, we give an algorithm that implements the leftmost augmenting-paths algorithm in at most $2\phi + 2$ phases. However, just as the analysis in Sections 3 doesn’t use Borradaile and Klein’s notion of unusability, neither does the algorithm in Appendix F. In Appendix F we more explicitly express the algorithm of Section 3 as an augmenting paths algorithm (and remove the extra space requirement). Since these algorithms are all variants or direct implementations of the leftmost augmenting paths algorithm, we should be able to appeal to the unusability of Borradaile and Klein to tighten the analysis. Combining these ideas may lead to algorithms that are both asymptotically and practically superior.

**Acknowledgements:** The authors thank Jérémie Barbay for very helpful discussions. This material is based upon work supported by the National Science Foundation under Grant No. CCF-0963921.
References

1. G. Borradaile and P. Klein. An $O(n \log n)$ algorithm for maximum st-flow in a directed planar graph. *J. ACM*, 56(2):1–30, 2009.
2. G. Borradaile, P. Klein, S. Mozes, Y. Nussbaum, and C. Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. In *Proc. FOCS*, pages 170–179, 2011.
3. Reinhard Diestel. *Graph Theory*. Springer-Verlag, 2005.
4. J. Erickson. Maximum flows and parametric shortest paths in planar graphs. In *Proc. SODA*, pages 794–804, 2010.
5. C. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.
6. R. Hassin. Maximum flow in $(s,t)$ planar networks. *IPL*, 13:107, 1981.
7. R. Hassin and D. B. Johnson. An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks. *SIAM J. Comp.*, 14:612–624, 1985.
8. M. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *JCSS*, 55(1):3–23, 1997.
9. Dorit S. Hochbaum. The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research*, 56(4):992–1009, 2008.
10. T. C. Hu. *Integer Programming and Network Flows*. Addison-Wesley, 1969.
11. A. Itai and Y. Shiloach. Maximum flow in planar networks. *SIAM J. Comp.*, 8:335–150, 1979.
12. D. B. Johnson and S. Venkatesan. Using divide and conquer to find flows in directed planar networks in $O(n^{3/2} \log n)$ time. In *Proc. Allerton*, pages 898–905, 1982.
13. Donald B. Johnson and Shankar M. Venkatesan. Partition of planar flow networks. In *Proc. FOCS*, pages 259–264, Washington, DC, USA, 1983. IEEE Computer Society.
14. H. Kaplan and Y. Nussbaum. Minimum st-cut in undirected planar graphs when the source and the sink are close. In *Proc. STACS*, pages 117–128, 2011.
15. Haim Kaplan and Yahav Nussbaum. Maximum flow in directed planar graphs with vertex capacities. In *Proc. ESA*, pages 397–407, 2009.
16. S. Khuller, J. Naor, and P. Klein. The lattice structure of flow in planar graphs. *SIAM J. Disc. Math.*, 6(3):477–490, 1993.
17. J. Reif. Minimum $s$-$t$ cut of a planar undirected network in $O(n \log^2 n)$ time. *SIAM J. Comp.*, 12:71–81, 1983.
18. D. Sleator and R. Tarjan. A data structure for dynamic trees. *JCSS*, 26(3):362–391, 1983.
19. E.H. Spanier. *Algebraic Topology*. Springer, 1994.
20. Robert E. Tarjan and Renato F. Werneck. Dynamic trees in practice. *J. Exp. Algorithmics*, 14:4.5–5.4.23, January 2010.
21. H. Whitney. Planar graphs. *Fundamenta mathematicae*, 21:73–84, 1933.
A  History of shortest-path based planar flow algorithms

| History of planar maximum flow and minimum cut via shortest sss |
|---------------------------------------------------------------|
| year problem | # SP calls | run time | reference                                      |
|---------------|------------|-----------|-----------------------------------------------|
| 1969 min cut, directed st-planar                            | 1          | $\Theta(n)$ | Hu [10]                                       |
| 1979 max flow, directed st-planar                            | 1          | $\Theta(n)$ | Itai and Shiloach [11]                        |
| 1983 max flow, directed                                     | $\phi$    | $\Theta(n\phi)$ | Johnson and Venkatesan [13]             |
| 1983 min cut, undirected                                     | $\leq n$   | $O(n\log n)$ | Reif [17]                                    |
| 1985 max flow, undirected                                     | $\leq n$   | $O(n\log n)$ | Hassin and Johnson [7]                        |
| 2011 min cut, undirected                                     | $\leq \phi$ | $O(n\log \phi)$ | Kaplan and Nussbaum [14]                     |
| 2012 max flow, directed                                     | $\leq 2\phi + 2$ | $O(n\phi)$ | [this paper, appendix]                         |
| 2012 max flow, directed                                     | 3          | $\Theta(n\phi)$ | [this paper]                                 |

Table 1. All run times are given using modern techniques, namely the linear-time shortest path [8] and the cycle-cancelling [15] algorithms, that post-date the results of last century. Note that the last algorithm has an $O(\phi)$ factor increase in required space.

B  Figures

(a) $P$ crosses $Q$ from right to left: $P$ enters $Q$ on the right at $x$, and $P$ leaves $Q$ on the left at $y$.
(b) $P$ and $Q$ are non-crossing.
(c) An illustration of the construction in the proof of Theorem [1]
(a) $G$ embedded on a cylinder with $s$-to-$t$ paths $P$ and $Q$.
(b) $G$ $\not\times P$ and $G$ $\not\times Q$.
(c) The covering graph $\tilde{G}$ of $G$.

\section*{C Proof of Lemma 9}

Let $C$ be the circulation that takes $f_{1}^{ST}$ to $f^{1}$. $C$ is leftmost w.r.t. $c_{f_{1}^{ST}}$, and by Lemma 4, can be decomposed into a set of clockwise flow cycles. Consider one such cycle $C$. This cycle is residual w.r.t. $c_{f_{1}^{ST}}$.

We first observe that $C$ must visit a vertex in the first $\phi$ copies or in the last $\phi$ copies of $G$ in $G^{ST}_{k}$.

If $C$ goes through $S$, $C$ contains consecutive darts $s^iS, Ss^i$. By the second part of Lemma 7 and the definition of $f_{1}^{ST}$, $Ss^i$ is saturated for all $i \geq \phi$. Therefore, $i < \phi$. If $C$ goes through $T$, by a similar argument, $C$ must visit a vertex in the last $\phi$ copies of $G$. If $C$ goes through neither $S$ nor $T$ and if $C$ contains no vertex in the first or last $\phi$ copies of $G$, then the darts of $C$ are residual in $G$ w.r.t. $f$ by the first part of Lemma 7. By Lemma 4, $C$ maps to a clockwise cycle in $G$, contradicting the leftmost-ness of $f$.

So, we consider the case when $C$ visits a vertex in the first $\phi$ copies of $G$; the case for the last $\phi$ copies is symmetric.

Suppose for a contradiction that $C$ contains a dart $d$ in a copy of $G$ greater than $2\phi + 1$. Then $C$ contains a path $P$ that starts in copy $\phi$ and ends in copy $2\phi + 1$ of $G$. By the first part of Lemma 7, the darts of $P$ are residual in $f_{1}$ and so they are also residual in $f$. By the pigeon-hole argument, $P$ contains two copies of the same vertex. By Lemma 4, $P$ contains a subpath that maps to a clockwise cycle in $G$. This contradicts that $f$ is leftmost. \qed
D  Leftmost maximum flows and shortest paths

Hassin’s algorithm starts by transforming the max flow problem into a max saturating circulation problem: it introduces a new infinite-capacity directed arc $ts$ embedded so that every $s$-to-$t$ residual path forms a clockwise cycle with $ts$ and then saturates the clockwise cycles. This saturation of clockwise cycles is computed via finding a shortest-paths tree rooted at $f^\infty$ in the dual graph $G^*$, interpreting primal capacities as dual distances. Let $\delta$ be the shortest-path distances in $G^*$ from $f^\infty$. Consider the flow:

$$f[d] = \delta[\text{head}_{G^*}(d)] - \delta[\text{tail}_{G^*}(d)] \forall \text{ darts } d.$$ (1)

After removing the artificial arc $ts$ from $G$, $f$ is a maximum feasible flow from $s$ to $t$ in $G$, and $\delta$ is the potential assignment that induces $f$.

Khuller, Naor and Klein [16] later showed that a flow derived from shortest-path distances in the dual in this manner is clockwise acyclic.

E  Convergence

We interpret the shortest-paths computation used in Section 3.3 as an augmenting paths algorithm in the original graph $G$, and illustrate that this will converge in a finite number of augmentations.

Consider the cover $G^{ST}$, as defined in Section 3, but with the restriction of $k$ copies removed, and let $f^{ST}$ be the leftmost maximum $ST$-flow in $G^{ST}$. By Lemma 9, the flow on the darts in copy $2\phi + 1$ of $G$ in $G^{ST}$ corresponds to the leftmost flow in $G$. If we relax the condition that the capacity of the arcs adjacent to $S$ and $T$ are finite, i.e. if we consider the leftmost maximum $ST$-flow $f^{ST}$ in $G^{ST}$ with capacities $c^{ST}$ instead of $c|f|$, we will need to consider a much later copy.

We show how to convert the flow $f_{1}^{ST}$ into the leftmost maximum flow $f^{ST}$, while not changing the flow on darts in copies $2\phi + 1$ or later except for saturating $\infty$-to-$T$ paths.

The construction follows much as that the proof of Lemma 9 except now we may have larger clockwise cycles through $T$. Such a cycle $C$ would be of the form $t^i T \circ T^i \circ P$, where $i < j$. $P$ corresponds to a counterclockwise residual cycle around $s$ in $G$, and so may be residual w.r.t. $f$. Beyond copy $2\phi + 1$, all copies of $C$ would be residual and their union would correspond to a set of $\infty$-to-$t$-to-$T$ paths. As such, the flow through $t^i$ may be greater than $|f|$. However, the from-$\infty$ paths will be saturated at some point: the amount of flow pushed along these paths can be no greater than $U$, the sum of the capacities of $G$. As this flow must be directed through some $t^i$, for $i > 4\phi + U$ (where the $4\phi$ is inherited from Lemma 9), the flow through $t^i$ will be $|f|$. Similarly, the flow in the copies of $G$ beyond $4\phi + U$ will be the same as $f$ except for the counterclockwise cycles

Since we always embed $t$ to be on $f^\infty$ in $G$, $f^\infty$ is the head of the dual dart $(ts)^*$ in $G^*$. 

around $s$ that participate in the $\infty$-to-$t$ paths. An example of this is illustrated in Figure 1.

This observation does not seem very useful algorithmically. However, consider how Dijkstra’s algorithm explores the dual of $G^ST$: when a face is popped off the priority queue, the wavefront of faces between those that have been popped and those that have not corresponds to an augmenting path. We can follow these augmenting paths in the original graph $G$. These augmenting paths push flow from $s$ to $t$, but when we are done with a path, we do not update the capacities, but simply continue. By the previous observation, this version of augmenting paths will converge to a maximum flow (in $O(n(U + \phi))$ iterations).

It is curious to note that although these augmentations correspond to leftmost augmentations, the flow we end up with may not be leftmost, as it may saturate a counterclockwise cycle around $s$, as in the example of Figure 1.

Fig. 1. (a) $G$ contains two counterclockwise cycles around $s$: $C_1$ and $C_2$, of capacities 1 and 2 respectively. Upon routing 1 unit of flow on the $s$-to-$t$ path, $C_1 \circ vt \circ tv$ and $C_2 \circ ut \circ tu$ are residual cycles through $t$ around $s$. (b) $G$’s half-infinite cover is given with grey dotted edges. Solid paths correspond to augmentations to $t_0$; the dashed path is an augmentation to $t_1$. Convergence occurs after $s_3$, instead of $s_0$.

F Maximum flow via sequential shortest-path computations

The algorithm implicit in Section 3.3 both requires the value of $\phi$ and will take time proportional to $\Theta(\phi)$. Here we present a different algorithm based on similar analysis techniques that is adaptive in the sense that the running time is naturally upper bounded by a function of $\phi$, but may have running time significantly less.

We require a finer understanding of the way in which paths may cross. We denote the sequence of crossings between $P$ and $Q$ by $P \otimes Q$ with ordering
inherited from $Q$. Although the ordering of $P \otimes Q$ and $Q \otimes P$ may not be the same, we have that $|P \otimes Q| = |Q \otimes P|$. While we only need part 2 of the following theorem, part 1 is used within the proof for part 2 and may be of independent interest.

**Theorem 2 (Leftmost Crossings).** Let $P$ be the leftmost residual $s$-to-$t$ path w.r.t. $c_{\otimes}$ and let $Q$ be an $s$-to-$t$ path such that rev($Q$) is residual w.r.t. $c_{\otimes}$. Then:

1. The order of crossings is the same along both $P$ and $Q$. That is, either $X = Y$ or $X = \text{rev}(Y)$ where $X$ and $Y$ are the $i$th crossing in $P \otimes Q$ and $Q \otimes P$, respectively.
2. $P$ crosses $Q$ from right to left at $X$ for all $X \in P \otimes Q$.

**Proof.** If $|P \otimes Q| = 0$, the theorem is trivially true.

Let $P \otimes Q = \{ X_1, X_2, \ldots, X_{|P \otimes Q|+1} \}$ and define $X_0 = s$, $X_{|P \otimes Q|+1} = t$. Likewise let $Q \otimes P = \{ Y_1, Y_2, \ldots, Y_{|Q \otimes P|} \}$ and define $Y_0 = s$, $Y_{|Q \otimes P|+1} = t$. For a contradiction to Part 1, let $i$ be the smallest index such that $X_i \notin \{ Y_j, \text{rev}(Y_j) \}$. Let $j$ be the index such that $Y_j \in \{ X_{i+1}, \text{rev}(X_{i+1}) \}$. Then $j \geq i$ by choice of $i$.

Let $x_i$ be any vertex in $X_i$. Since $P[x_{i-1}, x_{i+1}]$ does not cross $Q$ at $x_i$, $P[x_{i-1}, x_{i+1}]$ does not cross $Q[x_{i-1}, x_{i+1}]$. Since $P$ and rev($Q$) are residual, $C_1 = P[x_{i-1}, x_{i+1}] \circ \text{rev}(Q[x_{i-1}, x_{i+1}])$ is a simple counterclockwise cycle.

Since there are no crossings in $Q[x_i, x_{i+1}]$, $C_2 = Q[x_i, x_{i+1}] \circ P[x_{i+1}, x_i]$ is a simple cycle. Since $P$ is leftmost residual, $P[x_{i+1}, x_i]$ is left of $\text{rev}(Q[x_i, x_{i+1}])$ and $C_2$ is clockwise.

Since $P$ and $Q$ are simple, $C_1$ and $C_2$ do not cross. Therefore it must be the case that either $C_1$ is enclosed by $C_2$ or vice versa. See Fig. 2.

**C_2** is enclosed by **C_1** Since $P$ crosses $Q$ at $X_{i+1}$, $Q[x_{i+1}, \cdot]$ must have a subpath in the strict interior of $C_1$. Then, a maximal such subpath forms a counterclockwise cycle with a subpath of $P$, contradicting that $P$ is a leftmost residual path.

**C_1** is enclosed by **C_2** Since $P$ crosses $Q$ at $X_i$, $P[x_i, \cdot]$ must enter the strict interior of $C_1$. Since $P[x_i, \cdot]$ does not cross $Q[x_i, x_{i+1}]$, $P[x_i, \cdot]$ is entirely enclosed by $C_1$, contradicting that $t$ is on the infinite face.

This proves part 1 of the theorem. Since $Q[x_i, \cdot]$ does not cross $P[x_i, x_{i+1}]$, $Q[x_i, \cdot]$ does not enter the cycle $P[x_i, x_{i+1}] \circ \text{rev}(Q[x_i, x_{i+1}])$. Since $P[x_i, x_{i+1}]$ is right of $Q[x_i, x_{i+1}]$, it follows that $P$ enters $Q$ from the right at $x_{i+1}$. Part 2 follows.

**F.1 An adaptive algorithm**

In our new algorithm MAXADAPTIVEFLOW we find a sequence of capacities $c_0, c_1, \ldots$ where $c_i$ are the residual capacities of $c_{i-1}$ with respect to some flow. We refer to this flow as the flow that takes $c_{i-1}$ to $c_i$. 
MaxAdaptiveFlow \((G, s, t, c)\)

let \(c_0\) be the residual capacities resulting from saturating the c.w.
residual cycles of \(G\) w.r.t. \(c\)

\[ i := 0 \]

while there is a residual s-to-t path in \(G\) w.r.t. \(c_i\)

let \(A_i\) be the leftmost of these paths

let \(c_{i+1}\) be the residual capacities resulting from saturating the
leftmost st-flow in \(G \times A_i\) w.r.t. \(c_i\)

\[ i := i + 1 \]

return the flow defined by \(f[d] = c_i[d] - c[d] \forall\) darts \(d\).

Let \(\rho\) be the number of iterations of MaxAdaptiveFlow. \(A_i\) is an s-to-t
path, \(s\) and \(t\) are adjacent to \(f_\infty\) in \(G \times A_i\), and the leftmost st-flow in \(G \times A_i\) re-
duces to a single shortest-path computation in the dual graph. Therefore Max-
AdaptiveFlow can be implemented with \(\rho + 1\) shortest-path computations
(finding the initial circulation is an additional shortest-path computation). Max-
AdaptiveFlow is correct as it does not complete until there is no residual s-to-t
path. We will bound \(\rho\) by \(2\phi + 1\), giving:

**Theorem 3.** MaxAdaptiveFlow can be implemented using at most \(2\phi + 2\)
shortest-path computations.

We bound the number of iterations of MaxAdaptiveFlow by showing that
the paths \(A_0, A_1, \ldots\) first monotonically decrease and then increase in the number
of times they cross \(\Pi\), the smallest set of faces that a curve from \(s\) to \(t\) drawn
on the plane in which \(G\) is embedded must cross. Notice that \(A_i\) must cross
\(A_{i-1}\) at least once; since \(A_i\) is residual in \(G\) w.r.t. \(c_i\), \(A_i\) cannot be a path in
\(G \times A_{i-1}\) (for otherwise it would have been augmented in iteration \(i - 1\)). We show
that MaxAdaptiveFlow maintains as an invariant the absence of clockwise
residual cycles; this will imply that these crossings can only be from right to left.
(by Theorem 2). We relate the crossings between $A_i$ and $A_{i-1}$ to $A_i$ and $II$ by viewing these paths in the covering graph of $G$ (as defined in Section 3.1).

We could, in fact, show that MAXADAPTIVEFLOW implements the leftmost-augmenting-path algorithm of Borradaile and Klein [1] insofar as each iteration of MAXADAPTIVEFLOW corresponds to a consecutive batch of augmentations of the leftmost-augmenting-path algorithm. This correspondence would imply the following invariant. However, it is less cumbersome to prove this invariant directly.

**Invariant 1** $G$ has no clockwise residual cycles w.r.t. $c_i$.

**Proof (by induction).**

If $i = 0$ the invariant holds trivially by definition of $c_0$. For a contradiction, assume that $c_j$ is clockwise acyclic, but $c_{j+1}$ is not. Let $C$ be a clockwise cycle in $G$ that is residual w.r.t. $c_{j+1}$. Since $G \not\subseteq A_j$ is clockwise non-residual w.r.t. $c_{j+1}$, $C$ must use a dart $d$ of left $(A_j)$ that enters $A_j$. By the inductive hypothesis, $C$ is not residual w.r.t. $c_j$; let $a$ be its last non-residual arc w.r.t. $c_j$. Let $F$ be an $s$-to-$t$ path in the flow that takes $c_j$ to $c_{j+1}$ and uses $rev(a)$ and let $x$ be the first vertex of $F$ on $C$ after $head(a)$. Then: $F'[\cdot,x] \circ C[\cdot,head(d)]$ is residual w.r.t. $c_j$ and, since $F$ does not cross $A_j$, $F'[\cdot,x] \circ C[\cdot,head(d)] \circ rev(A_j)[head(d),s]$ is a clockwise cycle, which contradicts $A_j$ being leftmost residual w.r.t. $c_j$.

As a consequence of there being no clockwise residual cycles, we show that the paths $A_0, A_1, A_2, \ldots$ move from left to right.

**Lemma 10.** $A_i$ is left of $A_{i-1}$. $A_i$ crosses $A_{i-1}$ at least once and only from right to left.

**Proof.** First we observe that $rev(A_{i-1})$ is residual w.r.t. $c_i$. We find a leftmost maximum flow in $G \not\subseteq A_{i-1}$, an $st$-planar graph, and since $A_{i-1}$ is residual w.r.t. $c_{i-1}$, the flow we find is non-trivial. The leftmost of these flow paths must indeed by $A_{i-1}$.

Let $A_i \odot A_{i-1} = \{X_1, X_2, \ldots, X_{|A_i \odot A_{i-1}|}\}$ and define $X_0 = s, X_{|A_i \odot A_{i-1}|+1} = t$. Let $x_j$ be any vertex in $X_j$. Consider the subpaths $A_i[x_j, x_{j+1}]$ and $A_{i-1}[x_j, x_{j+1}]$. The cycle $A_i[x_j, x_{j+1}] \circ rev(A_{i-1}[x_j, x_{j+1}])$ is residual, and by Invariant 4 cannot be clockwise. Therefore $A_i[x_j, x_{j+1}]$ is left of $A_{i-1}[x_j, x_{j+1}]$.

**Proof (of Theorem 3).** Since $A_i$ crosses $A_{i-1}$ at least once and from right to left. Therefore there exists a subpath of $A_i$ such that $Y$ leaves $A_{i-1}$ from the left and enters $A_{i-1}$ from the right and there is no subpath of $A_i$ that leaves $A_{i-1}$ from the right and enters $A_{i-1}$ from the left. Then by Lemma 2 $A^0_i$ must contain at least one subpath from $A^i_{i-1}$ to $A_{i-1}'$ for some $j$ (and no $A^i_{i-1}$-to-$A_{i-1}'$ subpaths, for any $j$). Therefore we make progress in the following sense:

$A^0_i$ starts at a source strictly to the right of the source that $A^i_{i-1}$ starts at in $G$. 

Maximum $st$-flow in directed planar graphs via shortest paths
From the Pigeonhole Lemma 5, each $A^0_i$ can go through at most $\phi$ copies of $G$ in $G$. Therefore $A^0_i$ must start at a source within $\phi$ isomorphic copies of $G \times \Pi$ of the source that $I^{00}$ starts at (when $\pi$ is an artificial path as in the proof of the Pigeonhole Lemma). It follows that the number of iterations of $\text{MaxAdaptiveFlow}$ is at most $2\phi + 1$.

Note that the bound in the above argument is not tight. The progress could be much greater in each iteration of $\text{MaxAdaptiveFlow}$. \qed