Chapter 10

Single-source, single-sink max flow

10.1 Flow assignments, capacity assignments, and feasibility

For an arc vector $\gamma$ and a dart vector $c$, we may refer to $\gamma$ as a flow assignment and refer to $c$ as a capacity assignment, in which case we say $\gamma$ is capacity-respecting with respect to $c$ if $\gamma \leq c$, i.e. if every dart $d$ satisfies $\gamma[d] \leq c[d]$.

10.1.1 Negative capacities

Note that the values of $c$ are allowed to be negative. Let $\gamma$ be a flow assignment that is capacity-respecting with respect to $c$, and suppose $c[d] < -c$ for some dart $d$ and some positive number $c$. Then $\gamma[d] \leq -c$, so by antisymmetry $\gamma[\text{rev}(d)] \geq c$. Thus a negative capacity can be interpreted as a lower bound on the reverse dart.

10.2 Circulations

Recall from Section 3.3.5 that a vector in the cycle space of a graph $G$ is called a circulation in $G$. Let $c$ be a dart vector. Suppose $G$ is a connected planar embedded graph, and let $f_\infty$ be one of the faces. Recall from Section 4.4.1 that a circulation $\theta$ can be represented as a linear combination

$$\theta = \sum \{\rho[f] \eta(f) : f \in V(G^*)\}$$

(10.1)

or, more concisely, as $\theta = A_{G^*} \rho$, where $\rho$ is a vertex vector of $G^*$. We refer to the coefficients $\rho[f]$ as face potentials.
10.2.1 Capacity-respecting circulations in planar graphs

We consider the problem of computing a circulation that is capacity-respecting with respect to a given capacity assignment $c$. Because capacities can be negative, the all-zeroes circulation is not necessarily capacity-respecting. We show a link between face potentials in the primal and price vectors in the dual.

**Lemma 10.2.1.** Let $\theta$ be a circulation of $G$, and let $\rho$ be a price vector of $G^*$ such that $\theta = A_{G^*} \rho$. Then the circulation is capacity-respecting with respect to $c$ iff $\rho$ is a consistent price vector in $G^*$ with respect to $c$.

**Proof.** For every dart $d$,

$$c[d] - \theta[d] = c[d] - (\rho[\text{head}_{G^*}(d)] - \rho[\text{tail}_{G^*}(d)])$$

$$= c[d] - (c[d] - c_\rho[d])$$

$$= c_\rho[d]$$

Therefore the left-hand side is nonnegative iff the right-hand side is nonnegative. 

**Corollary 10.2.2** (Miller and Naor). Let $G$ be a connected plane graph, and let $c$ be a capacity function. There is a circulation in $G$ that is capacity-respecting with respect to $c$ iff the dual $G^*$ contains no cycle of darts whose length with respect to $c$ is negative.

**Proof.** (if) Suppose there is no negative-cost cycle. Let $\rho$ be a from-$f_\infty$ distance vector for some face $f_\infty$ of $G$. By Lemma 7.1.3, $\rho$ is a consistent price vector for $G^*$. By Lemma 10.2.1, $A_{G^*} \rho$ is a capacity-respecting circulation.

(only if) Suppose $\theta$ is a capacity-respecting circulation. As discussed in Section 4.4.1, there exists a face-potential vector $\rho$ such that $\theta = A_{G^*} \rho$. By Lemma 10.2.1, $\rho$ is a consistent price vector for $G^*$. By Lemma 7.1.5, $G^*$ has no negative-cost cycles.

10.3 $st$-flows

For nodes $s$ and $t$, a flow assignment $\gamma$ is an $st$-flow if it satisfies conservation at every node except possibly $s$ and $t$. The value of an $st$-flow $\gamma$ is

$$\sum \{\gamma[d] : \text{tail of } d \text{ is } s\}$$

One might be tempted to write this sum as a dot-product:

$$\eta(s) \cdot \gamma$$

However, the dot-product is the sum over all darts, including darts whose tails are $s$ (darts assigned +1 by $\eta(s)$) and darts whose heads are $s$ (darts assigned -1 by $\eta(s)$). Therefore the dot-product overcounts the value of $\gamma$ by a factor of two.
10.4. MAX LIMITED FLOW IN ST-PLANAR GRAPHS

**Inner product of arc vectors** For this reason, we define the inner product of two arc vectors as
\[
\langle \psi, \chi \rangle = \frac{1}{2}(\psi \cdot \chi)
\] (10.2)
With this definition, the value of an st-flow \( \gamma \) is \( \langle \eta(v), \gamma \rangle \).

**Max st-flow and max limited st-flow** Max st-flow is the following problem: given a graph \( G \) with a capacity assignment \( c \), and given vertices \( s, t \), compute an st-flow whose value is maximum. Ordinarily the capacity assignment is assumed to be nonnegative.

In a slight variant, L-limited max st-flow, one is additionally given a number \( L \), and the goal is to compute an st-flow whose value is maximum subject to the value being at most \( L \). If \( L \) is assigned an upper bound on the maximum value of an st-flow, e.g. the sum of capacities of the darts whose tail is \( s \), the limit has no effect. This shows that ordinary max-flow can be reduced to limited max-flow. (The reverse reduction is also not difficult.)

**Difference between st-flows**

**Lemma 10.3.1.** For two st-flows \( \phi_1 \) and \( \phi_2 \) with the same value, the difference \( \phi_1 - \phi_2 \) is a circulation.

**residual capacities** Given a capacity assignment \( c \) and an arc vector \( \gamma \), the residual capacity assignment \( c_{\gamma} \) is defined as
\[
c - \gamma
\]

**Lemma 10.3.2.** For any st-flow \( \gamma \) and capacity function \( c \), if \( \gamma' \) is an st-flow that satisfies the residual capacity function \( c_{\gamma} \), then \( \gamma + \gamma' \) satisfies the original capacity function \( c \).

**saturated darts** We say a dart \( d \) is saturated by \( \gamma \) with respect to \( c \) if \( \gamma[d] = c[d] \). Note that a dart that is saturated by \( \gamma \) with respect to \( c \) has zero residual capacity.

**Lemma 10.3.3.** For any st-flow \( \gamma \), if every s-to-t path contains a saturated dart then \( \gamma \) is a maximum st-flow.

10.4 Max limited flow in st-planar graphs

10.4.1 st-planar embedded graphs and augmented st-planar embedded graphs

A planar embedded graph \( G \) with an edge \( e \) having endpoints \( s \) and \( t \) is called an augmented st-planar embedded graph. The graph obtained from this graph by deleting \( e \) is called an st-planar graph. Equivalently, an st-planar embedded graph is a graph in which there is a face containing both \( s \) and \( t \). An st-planar graph is the underlying graph of an st-planar embedded graph.
10.4.2 The set-up

We consider the problem of finding a limited maximum st-flow in an st-planar graph with given nonnegative capacities. Let $G$ be an augmented st-planar graph, let $e$ be the edge with endpoints $s$ and $t$, let $c$ be the corresponding capacity assignment (where each dart of the added edge has zero capacity), and let $L$ be the limit.

Let $\hat{d}$ be the dart of $e$ with $\text{tail}_G(\hat{d}) = t$, and let $f_\infty = \text{tail}_{G^*}(\hat{d})$. Let $c'$ be a capacity assignment that differs from $c$ only in that $c'[\hat{d}] = L$.

From price vectors to circulations  Consider the function

$$\rho \mapsto A_{G^*}\rho$$

whose domain is the set of price vectors $\rho$ with $\rho[f_\infty] = 0$. Because the columns of $A_{G^*}$ (other than the column for $f_\infty$) form a basis for the cycle space of $G$, every circulation in $G$ is the image of exactly one such price vector. Therefore this function is a bijection from such price vectors to circulations.

Let us further restrict the domain to price vectors that are consistent with respect to $c'$. By Lemma 10.2.1, the function is still a bijection, and the codomain consists of circulations that are capacity-respecting with respect to $c'$.

From circulations to st-flows  Consider the function $\text{supp}_{s}(\theta)$ that maps a circulation $\theta$ to the flow assignment $\gamma$ such that

$$\gamma[d] = \begin{cases} 0 & \text{if } d \text{ is a dart of } e \\ \theta[d] & \text{otherwise} \end{cases}$$

Then $\gamma$ is an st-flow whose value equals $\theta[\hat{d}]$ (which equals $-\theta[\text{rev}(\hat{d})]$). Furthermore, any such st-flow is the image of some circulation. Therefore this function is a bijection from circulations to st-flows.

Let us further restrict the domain to circulations $\theta$ that are capacity-respecting with respect to $c'$. Then the function remains a bijection, and its codomain consists of st-flows that are capacity-respecting with respect to $c$ and have value at most $L$.

From price vectors to st-flows  By composing the two functions, we obtain a bijection

- from consistent price vectors $\rho$ with $\rho[f_\infty] = 0$

- to capacity-respecting st-flows with value at most $L$.

Furthermore, the value of the st-flow is $\rho[\text{head}_{G^*}(\hat{d})]$. 
Maximizing the value of the \(st\)-flow  Our goal, therefore, is to find a consistent price vector \(\mathbf{\rho}\) with \(\mathbf{\rho}[f_{\infty}] = 0\) that maximizes \(\mathbf{\rho}[\text{head}_{G^*}(d)]\). The following Lemma 10.4.1 shows that we can take \(\mathbf{\rho}\) to be the from-tail \(G^*\) distance vector with respect to \(c'\).

**Lemma 10.4.1.** Suppose \(\mathbf{\rho}\) is the from-\(r\) distance vector with respect to \(c\) for some vertex \(r\). For each vertex \(v\),

\[
\mathbf{\rho}[v] = \max \{ \gamma[v] : \gamma \text{ is a consistent price function such that } \gamma[r] = 0 \} \quad (10.3)
\]

**Proof.** Let \(T\) be an \(r\)-rooted shortest-path tree. The proof is by induction on the number of darts in \(T[v]\). The case \(v = r\) is trivial. For the induction step, let \(v\) be a vertex other than \(r\), and let \(d\) be the parent dart of \(v\) in \(T\), i.e. head\((d) = v\). Let \(u = \text{tail}(d)\). Let \(\gamma\) be the vertex vector attaining the maximum in 10.3.

By the inductive hypothesis,

\[
\mathbf{\rho}[u] \geq \gamma[u] \quad (10.4)
\]

Since \(\gamma\) is a consistent price function,

\[
\gamma[v] \leq \gamma[u] + c[d] \quad (10.5)
\]

By Lemma 7.1.3, \(\mathbf{\rho}\) is itself a consistent price function, so \(\gamma[v] \geq \mathbf{\rho}[v]\), and \(d\) is tight with respect to \(\mathbf{\rho}\), so

\[
\mathbf{\rho}[v] = \mathbf{\rho}[u] + c[d] \quad (10.6)
\]

By Equations 10.4, 10.5, and 10.6, we infer \(\mathbf{\rho}[v] \geq \gamma[v]\). We have proved \(\mathbf{\rho}[v] = \gamma[v]\). \(\square\)

**10.4.3 The algorithm**

Thus the algorithm for limited max \(st\)-flow is as follows.

1. Let \(\hat{d}\) be the dart of \(e\) whose tail is \(t\).
2. Let \(c'\) be the capacity assignment obtained from \(c\) by assigning capacity \(L\) to \(\hat{d}\).
3. Let \(\mathbf{\rho}\) be the from-tail \(G^*\) distance vector in \(G^*\) with respect to \(c'\).
4. Let \(\theta = A_{G^*}\mathbf{\rho}\).
5. Return the flow assignment \(\theta'\) obtained from \(\theta\) by setting \(\theta'[\hat{d}]\) and \(\theta'[\text{rev}(\hat{d})]\) equal to zero.

**Lemma 10.4.2.** The algorithm returns a max limited \(st\)-flow
10.5 Max flow in general planar graphs

10.6 The algorithm

The input to the algorithm is a planar embedded graph $G$, two nodes $s$ and $t$, and a capacity function $c$. We require that $c$ is nonnegative. (For traditional maximum flow, $c$ is positive on arcs and zero on the reverses of arcs.) The algorithm maintains an $st$-flow $\gamma$.

Let $f_\infty$ be a face incident to $t$. The algorithm calculates an $f_\infty$-rooted shortest-path tree $T^*$ in the dual $G^*$ using $c$ as a cost function. The algorithm represents $T^*$ by a table $\text{pred}[\cdot]$ giving, for each face $f$ other than $f_\infty$, the parent dart in $T^*$, i.e. the dart in $T^*$ whose head is $f$. The algorithm initializes $\gamma$ to be the circulation whose potential function is the function giving distances of faces in $T^*$. By Lemma 10.2.1, this circulation obeys capacities $c$. The algorithm builds a spanning tree $T$ of the primal $G$ using edges not represented in the dual shortest-path tree.

The algorithm then performs a sequence of iterations. In each iteration, the algorithm saturates the $s$-to-$t$ path in $T$, and then selects a saturated dart $\hat{d}$ from this path and ejects it from $T$. It ejects from $T^*$ the parent dart $d'$ of the head of $\hat{d}$ in $T^*$. It then inserts $\hat{d}$ into $T^*$ and inserts $\text{rev}(d')$ into $T$.

```python
def MaxFlow(G, c, s, t):
    let $T^*$ be an $f_\infty$-rooted shortest-path tree of darts in $G^*$ w.r.t. $c$
    for each vertex $f$ of $G^*$, assign $\text{pred}[f] :=$ the dart in $T^*$ whose head is $f$
    for each dart $d$,
    assign $\gamma[d] := \text{dist}_c(\text{head of } d \text{ in } G^*) - \text{dist}_c(\text{tail of } d \text{ in } G^*)$
    let $T$ be the tree formed by edges not represented in $T^*$
    while $t$ is reachable from $s$ in $T$:
        comment: push flow to saturate $s$-to-$t$ path in $T$
        let $P$ be the $s$-to-$t$ path in $T$
        $d := \text{arg min}\{c_\gamma(d) : d \in P\}$
        $\Delta := c_\gamma(d)$
        $\gamma[d] := \gamma[d] + \Delta$ for every dart $d$ in $P$
        $\gamma[(\text{rev}(d))] := \gamma[\text{rev}(d)] - \Delta$ for every dart $d$ in $P$
        let $q$ be the head of $\hat{d}$ in $G^*$
        eject $\hat{d}$ from $T$ and insert $\text{rev}(\text{pred}[q])$ into $T$
        $\text{pred}[q] := \hat{d}$ (comment: this replaces $\text{pred}[q]$ in $T^*$ with $\hat{d}$)
    return $\gamma$
```

We now state several invariants of the algorithm. The invariants all hold initially by construction; we prove that each iteration preserves them.
Invariant 10.6.1 (Borradaile and Klein). \( \gamma \) is a capacity-respecting \( st \)-flow with respect to \( c \).

Proof. By Lemma 10.2.1, initially \( \gamma \) is a circulation that respects the capacities. It is therefore a flow of value 0. In each iteration, some amount of flow is added to the darts comprising an \( s \)-to-\( t \) path. This preserves the property that \( \gamma \) is an \( st \)-flow. Since the amount added is no more than the minimum residual capacity of the darts in the path, after the flow is added every dart has nonnegative residual capacity.

Invariant 10.6.2 (Borradaile and Klein). An edge is represented in \( T \) iff it is not represented in \( T^* \).

Proof. Line 5 establishes the invariant, and Lines 12 and 13 preserve it.

Invariant 10.6.3 (Borradaile and Klein). The darts of the dual tree \( T^* \) are saturated by \( \gamma \).

Proof. First we show that the invariant holds initially. Let \( \rho \) be the vector of distances from \( f^\infty \) in \( G^* \) with respect to \( c \) in \( G^* \). Initially, for each dart \( d \),

\[
\gamma[d] = \rho[\text{head}_{G^*}(d)] - \rho[\text{tail}_{G^*}(d)]
\]

Since initially \( T^* \) is a shortest-path tree, each of its darts satisfies

\[
\rho[\text{head}_{G^*}(d)] = \rho[\text{tail}_{G^*}(d)] + c[d]
\]

which shows that \( \gamma[d] = 0 \).

Line 10 preserves the invariant because it does not affect darts of \( T^* \). To show that Line 13 preserves the invariant, note that after Line 10, the choice of \( \Delta \) ensures that \( c_\gamma[\hat{d}] = 0 \).

Corollary 10.6.4. Upon termination \( \gamma \) is a maximum \( st \)-flow that respects the capacities \( c \).

Proof. By Invariant 10.6.1, \( \gamma \) is an \( st \)-flow that respects the capacities \( c \). Upon termination, \( t \) is not reachable from \( s \) in \( T \). This means that there exists an \( st \)-cut, all of whose edges are not in \( T \). By Invariant 10.6.2, every edge of this cut is represented in \( T^* \), so the last dart \( \hat{d} \) inserted into \( T^* \) in Line 13 completed a cycle in \( T^* \). Therefore, just before the insertion, \( \text{head}_{G^*} (\hat{d}) \) must have been an ancestor of \( \text{tail}_{G^*} (\hat{d}) \) in \( T^* \). It follows that after the insertion \( T^* \) contains a cycle \( C \) of darts, such that \( \hat{d} \in C \). By invariant 10.6.3, all the darts of \( C \) are saturated by \( \gamma \). Since ejecting \( \hat{d} \) from \( T \) separated \( T \) into two connected components, such that \( s \) is in the component containing \( \text{tail}_{G^*} (\hat{d}) \), the darts of \( C \) form a saturated \( st \)-cut (rather than a \( ts \)-cut).
10.7 Erickson’s analysis

10.7.1 Dual tree is shortest-path tree

Invariant 10.7.1. $T^*$ is an $f_{\infty}$-rooted shortest-path tree with respect to the cost function $c - \gamma$.

Proof. By Invariant 10.6.1, every dart has nonnegative residual capacity, hence nonnegative cost. By Invariant 10.6.3, every dart in $T^*$ has zero residual capacity, hence zero cost. □

Invariant 10.7.2. Let $\gamma'$ be any st-flow whose value is the same as that of $\gamma$. Then $T^*$ is an $f_{\infty}$-rooted shortest-path tree with respect to the cost function $c - \gamma'$.

Proof. Since $\gamma$ and $\gamma'$ have the same value, $\gamma - \gamma'$ is a circulation, so there is a potential vector $\rho$ in vertex space such that $\gamma - \gamma' = A_{G^*} \rho$. Therefore $c - \gamma' = c - \gamma + \gamma = c - \gamma + A_{G^*} \rho$.

By Invariant 10.7.1, $T^*$ is a shortest-path tree with respect to $c - \gamma'$, so by Corollary 7.1.2, it is a shortest-path tree with respect to $c - \gamma'$. □

10.7.2 Crossing numbers

Let $P$ be an s-to-t path and let $D$ be a set of darts. The crossing number of $D$ with respect to $P$ is

$$\pi_P(D) = \langle \eta(P), \eta(D) \rangle$$

As we shall see, crossing numbers increase as the algorithm executes.

Lemma 10.7.3. Let $C$ be a directed cut. Let $P$ be an s-to-t path. Then $C$ is an s-t cut iff $\pi_P(C) = 1$.

To measure progress, we fix one s-to-t path $Q$ and consider crossing numbers with respect to $Q$.

Lemma 10.7.4 (Erickson). Every time $T^*[\text{head}_{G^*}(d)]$ changes for some dart $d$, $\pi_Q(T^*[\text{head}_{G^*}(d)])$ increases by 1.

Proof. Consider an iteration, and let $\hat{d}$ be the dart ejected from the primal tree $T$. Let $T^*$ be the dual tree before the iteration. Let $D$ be the set $\{\hat{d}\} \cup \{d : d$ a descendent of $\text{head}_{G^*}(\hat{d})$ in $T^*\}$. $D$ is the set of darts for which $T^*[\text{head}_{G^*}(d)]$ changes when $\hat{d}$ is inserted into $T^*$. Let $R = T^*[\text{head}_{G^*}(\hat{d})]$, and let $R' = T^*[\text{tail}_{G^*}(\hat{d})] \circ \hat{d}$. For every $d \in D$, the iteration replaces the prefix $R$ of $T^*[\text{head}_{G^*}(d)]$ with $R'$. It therefore suffices to show that $\pi_Q(R') = \pi_Q(R) + 1$.

Let $\hat{P}$ be the s-to-t path chosen in the iteration. Let $C$ be the fundamental cycle of $\hat{d}$ with respect to $T^*$ in $G^*$. The only edge represented both in $C$ and in $\hat{P}$ is the edge of $d$, and it is represented in the same direction in each,
so \( \langle \eta(\hat{P}), \eta(C) \rangle = 1 \). By Lemma 10.7.3, therefore, \( C \) is a directed cut in the primal. Again by Lemma 10.7.3, \( \langle \eta(Q), \eta(C) \rangle = 1 \). Since \( \eta(C) = \eta(R') - \eta(R) \), this completes the proof.

**Invariant 10.7.5** (Erickson). For any face \( f \), \( T^*[f] \) is the shortest \( f_\infty \)-to-\( f \) path with respect to \( c \) among all such paths with crossing number \( \pi_Q(T^*[f]) \).

**Proof.** Let \( \lambda \) be the value of the flow \( \gamma \). Let \( \gamma' = \lambda \eta(Q) \). Let \( i = \pi_Q(T^*[f]) \).

With respect to the cost function \( c - \gamma' \), any path \( P \) with crossing number \( i \) w.r.t. \( Q \) has cost

\[
(c - \gamma')(P) = c(P) - \lambda\langle \eta(Q), \eta(P) \rangle \\
= c(P) - \lambda i
\]

(10.7)

Let \( R \) be any \( f_\infty \)-to-\( f \) path such that \( \pi_Q(R) = i \). By Invariant 10.7.2,

\[
(c - \gamma')(R) \geq (c - \gamma')(T^*[f])
\]

so by two applications of (10.7),

\[
c(R) - \lambda i \geq c(T^*[f]) - \lambda i
\]

so \( c(R) \geq c(T^*[f]) \).

---

### 10.8 Covering space

Need to write about the notion of a covering space

#### 10.8.1 The Universal Cover

The **universal cover** of \( G = (A, \pi) \) with respect to a simple dual path \( Q \) is an infinite planar graph \( \overline{G} = (\overline{A}, \overline{\pi}) \), where \( \overline{A} = \{a_i : a \in A, i \in \mathbb{Z} \} \). For every orbit \( d^1, d^2, \ldots, d^k \) of \( \pi \) and every \( i \in \mathbb{Z} \), there is an orbit \( (\overline{d}^1, \overline{d}^2, \ldots, \overline{d}^k) \) of \( \overline{\pi} \), where

\[
\overline{d}_i = \begin{cases} 
  d_{i-1} & \text{if } \text{rev}(d) \in Q \\
  d_i & \text{otherwise}
\end{cases}
\]

Informally, \( \overline{G} \) can be constructed as follows. Consider an embedding of \( G \) on a the plane where \( \text{start}(Q) \) is the infinite face. Remove the faces corresponding to \( \text{start}(Q) \) and \( \text{end}(Q) \), and cut the embedding along \( Q \). Create an infinite sequence of copies \( G_i \) of the annuli thus obtained. For each vertex \( v \) of \( G \), there is a corresponding vertex \( v_i \) of \( G_i \) \( (v_i \) is called a *lift* of \( v \)). “Glue” consecutive copies \( G_{i-1} \) and \( G_i \) by edges \( u_{i-1}v_i \) for every edge \( uv \in Q \). See Figure 10.2. Edge costs in \( \overline{G} \) are inherited from \( G \).

There is a natural projection map \( \omega : \overline{G} \to G \) that maps \( v_i \) to \( v \). The preimage of any path \( P \) in \( G \) is an infinite set of paths in \( G^* \), which are called the *lifts* of \( P \). If \( \text{start}(P) = v \) and \( \text{end}(P) = u \), then for any integer \( i \) there is a
Figure 10.1: Cutting a graph $G$ along a dual path $Q$ and “opening up” the embedding.

Figure 10.2: The universal cover $\overline{G}$ of $G$ w.r.t. $Q$

lift of $P$ that starts at $v_i$, and ends at $u_{i + \pi_Q(P)}$. The faces $\text{start}(Q)$ and $\text{end}(Q)$ lift into two unbounded faces of $\overline{G}$. Every other face in $G$ lifts to an infinite sequence of faces in $\overline{G}$.

**Lemma 10.8.1.** Let $u,v$ be vertices of $G$. Let $P$ be a shortest $u$-to-$v$ path among all $u$-to-$v$ paths with crossing number $i$ w.r.t. $Q$. Then, for any $j$, $P$ is a projection of a shortest $u_{j-i}$-to-$v_j$ path in $\overline{G}$.

**Problem 10.1.** Prove Lemma 10.8.1.

### 10.9 Finishing the proof

To avoid clutter denote $f_{\infty}$ by $o$. For a vertex $r$ of $G^*$, let $P_\ell(r)$ denote $T^*[r]$ at the time that $\pi_Q(T^*[r])$ is $\ell$. Let $\overline{P}_\ell(r)$ denote the $o$-to-$r_0$ lift of $P_\ell(r)$ in $\overline{G}$.

**Lemma 10.9.1.** For every vertex $r \in G^*$, the paths $\{\overline{P}_\ell(r)\}_\ell$ are mutually noncrossing.

**Proof.** By Invariant 10.7.5, $P_\ell(r)$ is a shortest path among the $o$-to-$r$ paths whose crossing number is $\ell$. By Lemma 10.8.1, $\overline{P}_\ell(r)$ is a shortest path in $\overline{G}$. Since we assume shortest paths are unique, $\{\overline{P}_\ell(r)\}_\ell$ are mutually noncrossing since a crossing would give rise to two distinct shortest paths with the same endpoints.

**Theorem 10.9.2.** Each dart is ejected from $T^*$ at most once during the execution of the algorithm.
Proof. Consider the universal cover $\overline{G^*}$ of $G^*$ with respect to $Q$. Let $d = pq$ be a dart. Assume neither $d$ nor $\text{rev}(d)$ belong to $Q$ (the cases where $d \in Q$ or $\text{rev}(d) \in Q$ are handled in a similar manner).

Assume $d$ is ejected from $T^*$ twice. Let $i$ be $\pi_Q(T^*[q])$ just before the first time $d$ is ejected from $T^*$, and let $j$ be $\pi_Q(T^*[q])$ just before the second time $d$ is ejected from $T^*$. By Lemma 10.7.4, $i + 1 < j$. We have: $P_i(q) = P_i(p) \circ d$, $d \notin P_{i+1}(q)$, $P_j(q) = P_j(p) \circ d$, and $d \notin P_{j+1}(q)$.

By Invariant 10.7.5 and Lemma 10.8.1, $P_\ell(r)$ is a shortest path in $\overline{G^*}$ for any $\ell$ and $r$. Consider the cycle $C$ in $\overline{G^*}$ that consist of $P_i(p)$, $\text{rev}(P_j(p))$, and a $o_{-j} - to - o_{-i}$ path through the unbounded face $t^*$ of $\overline{G}$ ($t^*$ is the lift of $t$). See Figure 10.3. Since $P_i(q) = P_i(p) \circ d$, $p_0q_0$ and $P_j(q) = P_j(p) \circ d$, neither $d$ nor its reverse belong to $\overline{C}$. Consider $P_{i+1}(q)$. Since $o_{i+1}$ is enclosed by $\overline{C}$, and since $d \notin P_{i+1}(q)$, $q_0$ must be enclosed by $\overline{C}$, or else shortest paths must cross. Similarly, since $o_{j+1}$ is not enclosed by $\overline{C}$, and since $d \notin P_{j+1}(q)$, $q_0$ must not be enclosed by $\overline{C}$, a contradiction.

10.10 Efficient Implementation

The efficient implementation, is almost identical to that of the MSSP algorithm in Chapter 7. The roles of the primal and dual trees are swapped. The tree $T$ is represented by a dynamic tree that supports $\text{Link}$, $\text{Cut}$, $\text{Ancestor}$, $\text{FindMin}$, $\text{AddToAncestors}$, and $\text{Evert}$ in $O(\log n)$ (amortized) time. The tree $T^*$ is represented by a parent list. As was the case in Chapter 7, each step of the algorithm can be implemented in (amortized) $O(\log n)$ time, so the total execution time is $O(n \log n)$.

10.11 Chapter Notes

The maximum $st$-flow algorithm is due to Borradaile and Klein [Borradaile and Klein, 2006, ?]. Its analysis and two simplifications are due to Erickson [Erickson, 2010] (the simplifications were also suggested by Schmidt et al. [Schmidt et al., 2009]).