Chapter 16

Network Flows

CS 473: Fundamental Algorithms, Spring 2013
March 15, 2013

16.0.0.1 Everything flows

_Panta rei_ – everything flows (literally).
   Heraclitus (535–475 BC)

16.1 Network Flows: Introduction and Setup

16.1.0.2 Transportation/Road Network

16.1.0.3 Internet Backbone Network

16.1.0.4 Common Features of Flow Networks

(A) _Network_ represented by a (directed) graph \( G = (V, E) \).
(B) Each edge \( e \) has a _capacity_ \( c(e) \geq 0 \) that limits amount of _traffic_ on \( e \).
(C) _Source(s)_ of traffic/data.
(D) _Sink(s)_ of traffic/data.
(E) Traffic _flows_ from sources to sinks.
(F) Traffic is switched/interchanged at nodes. Flow abstract term to indicate stuff (traffic/data/etc) that flows from sources to sinks.

16.1.0.5 Single Source/Single Sink Flows

Simple setting:
(A) Single source s and single sink t.
(B) Every other node v is an internal node.
(C) Flow originates at s and terminates at t.

(A) Each edge e has a capacity \( c(e) \geq 0 \).
(B) Sometimes assume:
Source \( s \in V \) has no incoming edges, and sink \( t \in V \) has no outgoing edges.

Assumptions: All capacities are integer, and every vertex has at least one edge incident to it.

16.1.0.6 Definition of Flow

Two ways to define flows:
(A) edge based, or
(B) path based.
Essentially equivalent but have different uses.

Edge based definition is more compact.

16.1.0.7 Edge Based Definition of Flow

Definition 16.1.1. Flow in network \( G = (V, E) \), is function \( f : E \rightarrow \mathbb{R}^{\geq 0} \) s.t.
16.1.0.8 Flow...

Conservation of flow law is also known as *Kirchhoff’s law*.

16.1.0.9 More Definitions and Notation

Notation

(A) The inflow into a vertex \( v \) is \( f^{\text{in}}(v) = \sum_{e \text{ into } v} f(e) \) and the outflow is \( f^{\text{out}}(v) = \sum_{e \text{ out of } v} f(e) \).

(B) For a set of vertices \( A \), \( f^{\text{in}}(A) = \sum_{e \text{ into } A} f(e) \). Outflow \( f^{\text{out}}(A) \) is defined analogously.

Definition 16.1.2. For a network \( G = (V, E) \) with source \( s \), the value of flow \( f \) is defined as \( v(f) = f^{\text{out}}(s) - f^{\text{in}}(s) \).

16.1.0.10 A Path Based Definition of Flow

Intuition: Flow goes from source \( s \) to sink \( t \) along a path.

\( \mathcal{P} \): set of all paths from \( s \) to \( t \). \(|\mathcal{P}\)| can be exponential in \( n \).

Definition 16.1.3 (Flow by paths.). A flow in network \( G = (V, E) \), is function \( f : \mathcal{P} \to \mathbb{R}^\geq 0 \) s.t.

(A) **Capacity Constraint:** For each edge \( e \), total flow on \( e \) is \( \leq c(e) \).

\[ \sum_{p \in \mathcal{P} : e \in p} f(p) \leq c(e) \]

(B) **Conservation Constraint:** No need! Automatic.

Value of flow: \( \sum_{p \in \mathcal{P}} f(p) \).
16.1.0.11 Example

\[ P = \{p_1, p_2, p_3\} \]
\[ p_1 : s \rightarrow u \rightarrow t \]
\[ p_2 : s \rightarrow u \rightarrow v \rightarrow t \]
\[ p_3 : s \rightarrow v \rightarrow t \]
\[ f(p_1) = 10, f(p_2) = 4, f(p_3) = 6 \]

16.1.0.12 Path based flow implies edge based flow

**Lemma 16.1.4.** Given a path based flow \( f : \mathcal{P} \rightarrow \mathbb{R}_{\geq 0} \) there is an edge based flow \( f' : E \rightarrow \mathbb{R}_{\geq 0} \) of the same value.

**Proof:** For each edge \( e \) define \( f'(e) = \sum_{p \in \mathcal{P}} f(p) \).

**Exercise:** Verify capacity and conservation constraints for \( f' \).

**Exercise:** Verify that value of \( f \) and \( f' \) are equal

16.1.0.13 Example

\[ \mathcal{P} = \{p_1, p_2, p_3\} \]
\[ p_1 : s \rightarrow u \rightarrow t \]
\[ p_2 : s \rightarrow u \rightarrow v \rightarrow t \]
\[ p_3 : s \rightarrow v \rightarrow t \]
\[ f(p_1) = 10, f(p_2) = 4, f(p_3) = 6 \]
\[ f'(s \rightarrow u) = 14 \]
\[ f'(u \rightarrow v) = 4 \]
\[ f'(s \rightarrow v) = 6 \]
\[ f'(u \rightarrow t) = 10 \]
\[ f'(v \rightarrow t) = 10 \]
16.1.1 Flow Decomposition

16.1.1.1 Edge based flow to Path based Flow

Lemma 16.1.5. Given an edge based flow $f' : E \to \mathbb{R}^{\geq 0}$, there is a path based flow $f : \mathcal{P} \to \mathbb{R}^{\geq 0}$ of same value. Moreover, $f$ assigns non-negative flow to at most $m$ paths where $|E| = m$ and $|V| = n$. Given $f'$, the path based flow can be computed in $O(mn)$ time.

16.1.2 Flow Decomposition

16.1.2.1 Edge based flow to Path based Flow

Proof:[Proof Idea]
(A) Remove all edges with $f'(e) = 0$.
(B) Find a path $p$ from $s$ to $t$.
(C) Assign $f(p)$ to be $\min_{e \in p} f'(e)$.
(D) Reduce $f'(e)$ for all $e \in p$ by $f(p)$.
(E) Repeat until no path from $s$ to $t$.
(F) In each iteration at least one edge has flow reduced to zero.
(G) Hence, at most $m$ iterations. Can be implemented in $O(m(m + n))$ time. $O(mn)$ time requires care.

16.1.2.2 Example

16.1.2.3 Edge vs Path based Definitions of Flow

Edge based flows:
(A) compact representation, only $m$ values to be specified, and
(B) need to check flow conservation explicitly at each internal node.

Path flows:
(A) in some applications, paths more natural,
(B) not compact,
(C) no need to check flow conservation constraints.
Equivalence shows that we can go back and forth easily.

16.1.2.4 The Maximum-Flow Problem

Problem

Input A network $G$ with capacity $c$ and source $s$ and sink $t$.

Goal Find flow of maximum value.

Question: Given a flow network, what is an upper bound on the maximum flow between source and sink?

16.1.2.5 Cuts

Definition 16.1.6 (s-t cut). Given a flow network an s-t cut is a set of edges $E' \subset E$ such that removing $E'$ disconnects $s$ from $t$: in other words there is no directed $s \rightarrow t$ path in $E - E'$.

The capacity of a cut $E'$ is $c(E') = \sum_{e \in E'} c(e)$.

Caution:
(A) Cut may leave $t \rightarrow s$ paths!
(B) There might be many s-t cuts.

16.1.3 s – t cuts

16.1.3.1 A death by a thousand cuts
16.1.3.2 Minimal Cut

Definition 16.1.7 (Minimal s-t cut). Given a s-t flow network $G = (V, E)$, $E' \subseteq E$ is a minimal cut if for all $e \in E'$, if $E' \setminus \{e\}$ is not a cut.

Observation: given a cut $E'$, can check efficiently whether $E'$ is a minimal cut or not. How?

16.1.3.3 Cuts as Vertex Partitions

Let $A \subset V$ such that
(A) $s \in A$, $t \notin A$, and
(B) $B = V \setminus A$ (hence $t \in B$).

The cut $(A, B)$ is the set of edges

$$(A, B) = \{(u, v) \in E \mid u \in A, v \in B\}.$$  

Cut $(A, B)$ is set of edges leaving $A$.

Claim 16.1.8. $(A, B)$ is an s-t cut.

Proof: Let $P$ be any $s \rightarrow t$ path in $G$. Since $t$ is not in $A$, $P$ has to leave $A$ via some edge $(u, v)$ in $(A, B)$.

16.1.3.4 Cuts as Vertex Partitions

Lemma 16.1.9. Suppose $E'$ is an s-t cut. Then there is a cut $(A, B)$ such that $(A, B) \subseteq E'$.

Proof: $E'$ is an s-t cut implies no path from $s$ to $t$ in $(V, E - E')$.
(A) Let $A$ be set of all nodes reachable by $s$ in $(V, E - E')$.
(B) Since $E'$ is a cut, $t \notin A$.
(C) $(A, B) \subseteq E'$. Why? If some edge $(u, v) \in (A, B)$ is not in $E'$ then $v$ will be reachable by $s$ and should be in $A$, hence a contradiction.

Corollary 16.1.10. Every minimal s-t cut $E'$ is a cut of the form $(A, B)$.  


16.1.3.5 Minimum Cut

**Definition 16.1.11.** Given a flow network an s-t minimum cut is a cut $E'$ of smallest capacity amongst all s-t cuts.

**Observation:** exponential number of s-t cuts and no “easy” algorithm to find a minimum cut.

16.1.3.6 The Minimum-Cut Problem

**Problem**

**Input** A flow network $G$

**Goal** Find the capacity of a minimum s-t cut

16.1.3.7 Flows and Cuts

**Lemma 16.1.12.** For any s-t cut $E'$, maximum s-t flow $\leq$ capacity of $E'$.

**Proof**: Formal proof easier with path based definition of flow.

Suppose $f : \mathcal{P} \rightarrow \mathbb{R}_{\geq 0}$ is a max-flow. Every path $p \in \mathcal{P}$ contains an edge $e \in E'$. Why?

Assign each path $p \in \mathcal{P}$ to exactly one edge $e \in E'$. Let $\mathcal{P}_e$ be paths assigned to $e \in E'$. Then

$$v(f) = \sum_{p \in \mathcal{P}} f(p) = \sum_{e \in E'} \sum_{p \in \mathcal{P}_e} f(p) \leq \sum_{e \in E'} c(e).$$

16.1.3.8 Flows and Cuts

**Lemma 16.1.13.** For any s-t cut $E'$, maximum s-t flow $\leq$ capacity of $E'$.

**Corollary 16.1.14.** Maximum s-t flow $\leq$ minimum s-t cut.
16.1.3.9 Max-Flow Min-Cut Theorem

**Theorem 16.1.15.** In any flow network the maximum s-t flow is equal to the minimum s-t cut.

Can compute minimum-cut from maximum flow and vice-versa!
Proof coming shortly.
Many applications:
(A) optimization
(B) graph theory
(C) combinatorics

16.1.3.10 The Maximum-Flow Problem

Problem

**Input** A network $G$ with capacity $c$ and source $s$ and sink $t$.

**Goal** Find flow of maximum value from $s$ to $t$.

**Exercise:** Given $G, s, t$ as above, show that one can remove all edges into $s$ and all edges out of $t$ without affecting the flow value between $s$ and $t$. 