Previously we studied graph algorithms on undirected graphs and examined algorithms for finding minimum spanning trees and all pairs shortest paths. The weights assigned to edges were often interpreted as distances.

Here we will consider directed graphs with weights on directed edges, and often it is convenient for these weights to correspond to the capacity of a pipe, conduit, or transportation channel. The graph can be interpreted as a flow network mapping possible distribution pathways for moving material. Commonly there is a single source $s$ where material is produced and a single sink (or target) $t$ where it is consumed. The material can be current, goods and supplies, fluid, information, passengers, etc. Often we want to find a way to move the maximum flow from $s$ to $t$ through a potentially branching set of connections. This is the maximum flow problem.
Flow network: directed graph $G = (V, E)$ with source vertex $s$ and sink $t$; each edge $e (u, v) \in E$ has non-negative capacity $c(u, v)$; if $E$ contains $(u, v)$ then it doesn't contain $(v, u)$ [the reverse edge]; if $(u, v) \not\in E$ then $c(u, v) = 0$.

- Also no self loops
- Assume each vertex lies on a path from source to sink - connected

Flow in $G$: function $f: V \times V \to \mathbb{R}$ satisfying

1. Capacity constraint: $0 \leq f(u, v) \leq c(u, v) \quad \forall u, v \in V$
2. Flow conservation: $\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) \quad \forall u \in V - \{s, t\}$

Flow Value: $|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$

It can be shown that this is the total net flow entering sink.
Goal: Find Maximum Flow Values

Residual Capacity of edge $(u,v)$: Given flow network $G=(V,E)$ and flow $f$ in $G$

$$ f_{(u,v)} = \begin{cases} 
(c_{(u,v)} - f_{(u,v)}) & \text{if } (u,v) \in E \\
0 & \text{ otherwise}
\end{cases} $$

- excess capacity
- amount $f_{(v,u)}$ can be decreased by

Residual Network $G_f$ is graph of edges with strictly positive residual capacity (that can support more flow),

- similar to Flow Network
- except can contain an edge and its reverse

$|E_f| \leq 2|E|$

Augmenting Path is a path from $S$ to $T$ in $G_f$

Because the residual network $G_f$ contains only edges that have excess capacity, any augmenting path in $G_f$ represents a perturbation of the flow $f$ with increased flow value.

Flow value can be increased along augmenting path $p$ by

$$ c_f(p) = \min_{(u,v) \in p} \{ f_{(u,v)} \} $$
1. Start with zero flow
2. Find an augmenting path in corresponding $G_f$
3. Augment $f$ by $G_f(p)$ and loop

Augmenting Flow

$$(f \rightarrow f') (u,v) = \begin{cases} 
  f(u,v) + f'(u,v) - f'(v,u) & \text{if } (u,v) \in E \\
  0 & \text{otherwise}
\end{cases}$$

Flow/capacity

Augmented flow

residual network

residual network

no augmenting path possible
Lemma 26.1  If \( f \) is a flow in \( G=(V,E) \) and \( f' \) is a flow in the corresponding residual network \( G_f \), then the augmentation of flow \( f \) by \( f' \) has value given by the sum of the flow values of \( f \) and \( f' \):

\[ |f \oplus f'| = |f| + |f'| \]

- Proof in book
  - first show \( f \oplus f' \) obeys capacity constraint for each edge in \( E \) and flow conservation for each vertex in \( V - \{s,t\} \)
  - then expand summation in definition of \( |f \oplus f'| \) to show equality

Lemma 26.2  Let \( f \) be a flow in \( G=(V,E) \) and \( p \) an augmenting path in \( G_f \). Define \( f_p : V \times V \rightarrow \mathbb{R} \) by

\[
  f_p(u,v) = \begin{cases} 
    Cf(p) & \text{if } (u,v) \text{ is on } p \\
    0 & \text{otherwise}
  \end{cases}
\]

Then \( f_p \) is a flow in \( G_f \) with value \( |f_p| = Cf(p) > 0 \)

- Corollary 26.3  Let \( f \) be a flow in \( G=(V,E) \) and \( p \) an augmenting path in \( G_f \) with a flow \( f_p \) defined as above. Then \( f \oplus f_p \) is a flow in \( G \) with value \( |f \oplus f_p| = |f| + |f_p| > |f| \)

Cuts of flow networks  A cut \( (S,T) \) of flow network \( G=(V,E) \) is a partition of \( V \) into \( S \) and \( T=V-S \) with \( s \in S \) and \( t \in T \).

For flow \( f \), the net flow \( f(S,T) \) across the cut \( (S,T) \) is defined as

\[
f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{v \in S} \sum_{v \in T} f(v,u)
\]
Capacity of cut \( (S,T) \)
\[
c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v)
\]

A minimum cut of a network is a cut whose capacity is minimum over all cuts of the network.

\[
f(S,T) = 12 + 11 - 4 = 19
\]
\[
c(S,T) = 12 + 14 = 26
\]

Note that flow across a cut takes into account both directions, because it is the actual net flow. Capacity only uses the "positive" direction because it represents a potential upper bound on flow.

Lemma 26.4: For \( f \) being a network flow in \( G \) with source \( s \) and sink \( t \) and for any cut of \( G \) \( (S,T) \), then the net flow across \( (S,T) \) is \( f(S,T) = |f| \).

Net flow across cut is the value of the flow in the network.
Corollary 26.5: The value of any flow \( f \) in flow network \( G \) is bounded from above by the capacity of any cut of \( G \).

Proof:

\[
|f| = f(s, t) \quad \text{Lemma 26.4}
\]

\[
= \sum \sum f(u, v) - \sum \sum f(v, u) \quad \text{definition of flow across a cut}
\]

\[
\leq \sum \sum f(u, v) \quad \text{flows are non-negative}
\]

\[
\leq \sum \sum c(u, v) \quad \text{capacity constraint}
\]

\[
= c(s, t)
\]

A consequence of this corollary is that the value of a maximum flow is bounded from above by the capacity of a minimum cut \( \rightarrow \) in fact, they are equal.

**Max-Flow Min-Cut Theorem**

For flow \( f \) in \( G=(V,E) \) with sources \( s \) and sink \( t \), the following are equivalent:

1. \( f \) is a maximum flow in \( G \)
2. The residual network \( G_f \) contains no augmenting paths
3. \( |f| = c(s, t) \) for some cut \((S,T)\) of \( G \)
**Proof Sketch**

1. $\Rightarrow 2$. Augmenting path would allow flow in $G$ to be increased above $|f|$, contradicting it being a maximum flow.

2. $\Rightarrow 3$. By corollary 26.5, $|f|$ is bounded from above by the capacity of any cut, so $|f| = c(S, \bar{T})$ means it can't be increased.

3. $\Rightarrow 1$. $G_f$ has no augmenting path $(s \rightarrow t)$. Let $S = \{v \in V : \exists \text{ path } s \rightarrow v \text{ in } G_f\}$ and $T = V - S$. Partition $(S, T)$ is a cut ($s \in S$, $t \in T$ because no path $s \rightarrow t$). Can show this is a minimum cut with $c(S, T) = |f|$.

**Ford-Fulkerson (G, s, t)**

- for each edge $(u, v) \in G, E$ 
  - $(u, v).f = 0$
  - while exists path $p$ $s \rightarrow t$ in $G_f$
  - $C_f(p) = \min \{C_f(u, v) : (u, v) \text{ is in } p\}$
    - for each edge $(u, v)$ in $p$
      - if $(u, v) \in E$
        - $(u, v).f = (u, v).f + C_f(p)$
      - else $(v, u).f = (v, u).f - C_f(p)$

` Initialize flow $f$ to zero

← seek augmenting path

- augment flow $f$ along $p$ by residual capacity $C_f(p)$

- each edge in $p$ is either also in $G$ or its reversal is in $G$ and flow is augmented appropriately

- when no augmenting flow can be found, $f$ is maximum flow.
Imagine we restrict ourselves to integer capacities, which frequently happens in practice or can be arranged by a transformation. If \( f^* \) is the maximum flow, then the while loop executes at most \( |f^*| \) times because each augmenting path increases flow by at least a unit integer.

Imagine data structure \( G' = (V, E') \) where \( E' \) contains each edge in \( G \) plus its reversal. With appropriate attributes can store capacities and flows in \( G \) and in \( G' \).

Augmenting paths can be found in residual network by breadth-first or depth-first search in \( O(V + E') = O(E) \) time.

Thus, total running time \( O(E |f^*|) \)

**Possible Efficiency Problem:**

 augmentation path with capacity 1

 augmentation path with capacity 1

... could take 2,000,000 augmentations to find more than...
Edmonds-Karp Algorithm

Improvement of Ford-Fulkerson using breadth-first search to find augmenting path as shortest path start in Gf, but with each edge having unit distance (weight).

Total number of augmentations is $O(VE)$ and total time is $O(VE^2)$

- removes dependence on $|f|^4$

Proof is in book - check it out - sketch here

- first shows shortest path distance increases monotonically with each flow augmentation

- an edge along an augmenting path is critical (the bottleneck) at most $O(v)$ times

- $O(VE)$ pairs of vertices can be edges in Gf and each can be critical $O(v)$ times $\Rightarrow O(VE)$

Further improvements in book (not responsible for)

$\Rightarrow O(V^2E) \Rightarrow O(V^3)$
A few issues:

1. What happens if flow network of actual problem has antiparalleled edges?

2. Networks with multiple sources and sinks