Definitions

Input:
- Directed Graph $G = (V, E)$
- Capacities $C(u, v) > 0$ for $(u, v) \in E$ and $C(u, v) = 0$ for $(u, v) \notin E$
- A source node $s$, and sink node $t$
Capacity

\[
\begin{align*}
v_1 & \rightarrow v_2 & 12 \\
v_3 & \rightarrow v_2 & 20 \\
s & \rightarrow v_1 & 16 \\
v_3 & \rightarrow s & 13 \\
v_3 & \rightarrow v_4 & 14 \\
v_4 & \rightarrow t & 4 \\
v_2 & \rightarrow t & 7 \\
v_1 & \rightarrow v_3 & 4 \\
v_2 & \rightarrow v_3 & 9 \\
v_1 & \rightarrow v_4 & 10 \\
v_4 & \rightarrow v_1 & 12 \\
\end{align*}
\]
Definitions

Input:

- Directed Graph $G = (V, E)$
- Capacities $C(u, v) > 0$ for $(u, v) \in E$ and $C(u, v) = 0$ for $(u, v) \notin E$
- A source node $s$, and sink node $t$

Output: A flow $f$ from $s$ to $t$ where $f : V \times V \to \mathbb{R}$ satisfies

- Skew-symmetry: $\forall u, v \in V, f(u, v) = -f(v, u)$
- Conservation of Flow: $\forall v \in V - \{s, t\}, \sum_{u \in V} f(u, v) = 0$
- Capacity Constraints: $\forall u, v \in V, f(u, v) \leq C(u, v)$

Goal: Maximize "size of the flow", i.e., the total flow coming leaving $s$:

$$|f| = \sum_{v \in V} f(s, v)$$
Capacity

The diagram shows a network with nodes labeled as $s$, $v_1$, $v_2$, $v_3$, and $t$. The edges are labeled with capacities.

- The edge from $s$ to $v_1$ has a capacity of 16.
- The edge from $v_1$ to $v_2$ has a capacity of 12.
- The edge from $v_2$ to $t$ has a capacity of 20.
- The edge from $s$ to $v_3$ has a capacity of 13.
- The edge from $v_3$ to $v_1$ has a capacity of 10.
- The edge from $v_3$ to $v_4$ has a capacity of 14.
- The edge from $v_4$ to $v_2$ has a capacity of 9.
- The edge from $v_4$ to $t$ has a capacity of 7.

The capacities of the edges are indicated in the diagram.
Capacity/Flow

Graph with nodes labeled as follows:

- Source: $s$
- Intermediate nodes: $v_1$, $v_3$, and $v_4$
- Sink: $t$

Edges and their capacities:

- $s$ to $v_1$: $16/11$
- $v_1$ to $v_2$: $12/12$
- $v_1$ to $v_3$: $10/0$
- $v_3$ to $v_2$: $9/4$
- $v_2$ to $v_4$: $20/15$
- $v_2$ to $t$: $7/7$
- $v_3$ to $t$: $4/4$
- $v_3$ to $v_4$: $14/11$

The flow through the network is balanced, as indicated by the capacities and flows on each edge.
Cut Definitions

Definition
An \( s - t \) cut of \( G \) is a partition of the vertices into two sets \( A \) and \( B \) such that \( s \in A \) and \( t \in B \).

Definition
The capacity of a cut \((A, B)\) is

\[
C(A, B) = \sum_{u \in A, v \in B} C(u, v)
\]

Definition
The flow across a cut \((A, B)\) is

\[
f(A, B) = \sum_{u \in A, v \in B} f(u, v)
\]

Note that because of capacity constraints: \( f(A, B) \leq C(A, B) \)
First Cut

\begin{center}
\begin{tikzpicture}
  \node[shape=circle,draw=black] (s) at (0,0) {$s$};
  \node[shape=circle,draw=black] (v1) at (2,2) {$v_1$};
  \node[shape=circle,draw=black] (v2) at (4,4) {$v_2$};
  \node[shape=circle,draw=black] (v3) at (2,-2) {$v_3$};
  \node[shape=circle,draw=black] (v4) at (4,-4) {$v_4$};
  \node[shape=circle,draw=black] (t) at (6,0) {$t$};

  \path (s) edge node [above] {$16/11$} (v1);
  \path (s) edge node [below] {$13/8$} (v3);
  \path (v1) edge node [above] {$10/0$} (v3);
  \path (v1) edge node [above] {$4/1$} (v4);
  \path (v2) edge node [above] {$12/12$} (v1);
  \path (v2) edge node [below] {$20/15$} (t);
  \path (v3) edge node [below] {$7/7$} (v4);
  \path (v4) edge node [below] {$4/4$} (t);
\end{tikzpicture}
\end{center}
Second Cut
Lemma

For any flow \( f \): for all \( s-t \) cuts \((A, B)\), \( f(A, B) \) equals \(|f|\).

Proof.

- By induction on size of \( A \) where \( s \in A \)
- **Base Case:** \( A = \{s\} \) and \( f(s, V - s) = |f| \)
- **Induction Hypothesis:** \( f(A, B) = |f| \) for all \( A \) such that \(|A| = k\)
- Consider cut \((A', B')\) where \(|A'| = k + 1\). Let \( u \in A' - s:\)

\[
f(A', B') = f(A' - u, B' + u) - \sum_{v \in A'} f(v, u) + \sum_{v \in B'} f(u, v)
\]

- By skew-symmetry and conservation of flow

\[
\sum_{v \in A'} f(v, u) - \sum_{v \in B'} f(u, v) = \sum_{v \in A'} f(v, u) + \sum_{v \in B'} f(v, u) = \sum_{v \in V} f(v, u) = 0
\]

- Hence, \( f(A', B') = f(A' - u, B' + u) = |f| \) by induction hypothesis.
Theorem (Max-Flow Min-Cut)

For any flow network and flow $f$, the following statements are equivalent:

1. $f$ is a maximum flow.
2. There exists an $s-t$ cut $(A, B)$ such that $|f| = C(A, B)$
Residual Networks and Augmenting Paths

Residual network encodes how you can change the flow between two nodes given the current flow and the capacity constraints.

**Definition**

Given a flow network $G = (V, E)$ and flow $f$ in $G$, the residual network $G_f$ is defined as

$$G_f = (V, E_f) \text{ where } E_f = \{(u, v) : C(u, v) - f(u, v) > 0\}$$

$$C_f(u, v) = C(u, v) - f(u, v)$$

Note that $(u, v) \in E_f$ implies either $C(u, v) > 0$ or $C(v, u) > 0$.

**Definition**

An augmenting path for flow $f$ is a path from $s$ to $t$ in graph $G_f$. The bottleneck capacity $b(p)$ is the minimum capacity in $G_f$ of any edge of $p$. We can increase flow by $b(p)$ along an augmenting path.
Capacity/Flow
Residual
Augmenting Path
Old Flow

\begin{figure}
\centering
\begin{tikzpicture}
  \node (s) at (0,0) {$s$};
  \node (v1) at (2,2) {$v_1$};
  \node (v2) at (4,2) {$v_2$};
  \node (v3) at (2,-2) {$v_3$};
  \node (v4) at (4,-2) {$v_4$};
  \node (t) at (6,0) {$t$};

  \draw[->] (s) -- (v1) node [midway, above] {$16/11$};
  \draw[->] (v1) -- (v2) node [midway, above] {$12/12$};
  \draw[->] (v2) -- (t) node [midway, above] {$20/15$};
  \draw[->] (s) -- (v3) node [midway, above] {$10/0$};
  \draw[->] (v3) -- (v4) node [midway, above] {$14/11$};
  \draw[->] (v4) -- (t) node [midway, above] {$4/4$};
  \draw[<->] (s) -- (v3) node [midway, above] {$13/8$};
  \draw[<->] (v1) -- (v3) node [midway, above] {$4/1$};
  \draw[<->] (v2) -- (v4) node [midway, above] {$7/7$};
  \draw[<->] (v1) -- (v4) node [midway, above] {$9/4$};
\end{tikzpicture}
\end{figure}
New Flow
Min Capacity Cut Proves this is Optimal
Old Residual Graph
New Residual Graph
Max-Flow Min-Cut

Theorem (Max-Flow Min-Cut)

For any flow network and flow $f$, the following statements are equivalent:

1. $f$ is a maximum flow.
2. There exists an $s - t$ cut $(A, B)$ with $|f| = f(A, B) = C(A, B)$.
3. There doesn’t exist an augmenting path in $G_f$.

Proof.

- $(2 \Rightarrow 1)$: Increasing flow, increases $f(A, B)$ which violates capacity
- $(1 \Rightarrow 3)$: If $p$ is an augmenting path, can increase flow by $b(p)$
- $(3 \Rightarrow 2)$: Suppose $G_f$ has no augmenting path. Define cut

$$A = \{ v : v \text{ is reachable from } s \text{ in } G_f \} \text{ and } B = V - A$$

$$\forall u \in A, v \in B, f(u, v) = C(u, v). \text{ Hence } C(A, B) = f(A, B) = |f|$$
Ford-Fulkerson Algorithm

Algorithm
1. \( \text{flow } f = 0 \)
2. while there exists an augmenting path \( p \) for \( f \)
   2.1 find augmenting path \( p \)
   2.2 augment \( f \) by \( b(p) \) units along \( p \)
3. return \( f \)

Theorem
The algorithm finds a maximum flow in time \( O(|E| f^*|) \) if capacities are integral where \( |f^*| \) is the size of the maximum flow.

Proof.
\( O(|E|) \) time to find each augmenting path via BFS and \( |f^*| \) iterations because each augmenting path increases flow by at least 1.
Ford-Fulkerson Algorithm with Edmonds-Karp Heuristic

**Algorithm**

1. \( flow \ f = 0 \)
2. while there exists an augmenting path \( p \) for \( f \)
   2.1 find shortest (unweighted) augmenting path \( p \)
   2.2 augment \( f \) by \( b(p) \) units along \( p \)
3. return \( f \)

**Theorem**

The algorithms finds a maximum flow in time \( O(|E|^2|V|) \)
Proof of Running Time (1/3)

Definition
Let $\delta_f(s, u)$ be length of shortest unweighted path from $s$ to $u$ in the $G_f$.

Definition
$(u, v)$ is critical if it’s on augmenting path $p$ for $f$ and $C_f(u, v) = b(p)$.

Lemma
$\delta_f(s, v)$ is non-decreasing as $f$ changes.

Lemma
Between occasions when $(u, v)$ is critical, $\delta_f(s, u)$ increases by at least 2.

Proof of Running Time.

- Max distance in $G_f$ is $|V|$ so any edge is critical at most $|V|/2$ times
- At most $2|E|$ edges in residual network
- There’s a critical edge in each iteration so $O(|E||V|)$ iterations
- Each iteration takes $O(|E|)$ to find shortest path
Proof of Running Time (2/3)

Lemma
\( \delta_f(s, v) \) is non-decreasing as \( f \) changes.

Proof.

- Consider augmenting \( f \) to \( f' \)
- For contradiction, pick \( v \) that minimizes \( \delta_{f'}(s, v) \) subject to:
  \[
  \delta_{f'}(s, v) < \delta_f(s, v)
  \]
  and let \( u \) be vertex before \( v \) on shortest path in \( G_{f'} \) from \( s \) to \( v \)
- Claim \((u, v) \notin E_f\)
  - Otherwise \( \delta_f(s, v) \leq \delta_f(s, u) + 1 \)
  - But \( \delta_f(s, u) \leq \delta_{f'}(s, u) \) and so \( \delta_f(s, v) \leq \delta_{f'}(s, u) + 1 = \delta_{f'}(s, v) \)
- \((u, v) \notin E_f \) and \((u, v) \in E_{f'}\) implies augmentation contains \((v, u)\)
- Since augmentation was shortest path:
  \[
  \delta_f(s, v) = \delta_f(s, u) - 1 \leq \delta_{f'}(s, u) - 1 = \delta_{f'}(s, v) - 2
  \]
Lemma

Between occasions when \((u, v)\) is critical, \(\delta_f(s, u)\) increases by at least 2.

Proof.

- Let \((u, v)\) be critical in the augmentation of \(f\)
- Since \((u, v)\) on shortest path: \(\delta_f(s, u) = \delta_f(s, v) - 1\)
- After augmentation \((u, v)\) disappears from residual network!
- Let \(f''\) be next flow with \((u, v) \in G_{f''}\) and \(f'\) be flow right before \(f''\)
- \((u, v) \notin G_f\) but \((u, v) \in G_{f''}\) implies \((v, u)\) used to augment \(f'\)
- Therefore \(\delta_{f'}(s, v) = \delta_{f'}(s, u) - 1\) and so

\[
\delta_f(s, u) = \delta_f(s, v) - 1 \leq \delta_{f'}(s, v) - 1 = \delta_{f'}(s, u) - 2
\]