6.6: Maximum flow

Frank Stajano  Thomas Sauerwald
Introduction

Ford-Fulkerson

Max-Flow Min-Cut Theorem
History of the Maximum Flow Problem [Harris, Ross (1955)]

Maximum Flow is 163,000 tons per day!
Maximum Flow is 163,000 tons per day!
Flow Network

- Abstraction for material (one commodity!) flowing through the edges
- \( G = (V, E) \) directed graph without parallel edges
- distinguished nodes: source \( s \) and sink \( t \)
- every edge \( e \) has a capacity \( c(e) \)

How to find a Maximum Flow?
Flow Network

- Abstraction for material (one commodity!) flowing through the edges
- \( G = (V, E) \) directed graph without parallel edges
- distinguished nodes: source \( s \) and sink \( t \)
- every edge \( e \) has a capacity \( c(e) \)

Capacity function \( c : V \times V \rightarrow \mathbb{R}^+ \)

How to find a Maximum Flow?

6.6: Maximum Flow
Flow Network

- Abstraction for material (one commodity!) flowing through the edges
- $G = (V, E)$ directed graph without parallel edges
- distinguished nodes: source $s$ and sink $t$
- every edge $e$ has a capacity $c(e)$

Capacity function $c : V \times V \rightarrow \mathbb{R}^+$

$c(u, v) = 0 \iff (u, v) \notin E$

How to find a Maximum Flow?

6.6: Maximum flow
A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

1. For every $u, v \in V$, $f(u, v) \leq c(u, v)$.
2. For every $v \in V \setminus \{s, t\}$, $\sum_{u \in V} f(u, v) = \sum_{v \in V} f(v, u)$.
3. For every $u, v \in V$, $f(u, v) = -f(v, u)$.

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$.
A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$.

How to find a Maximum Flow:

6.6: Maximum flow T.S.
Flow Network

A flow is a function \( f : V \times V \rightarrow \mathbb{R} \) that satisfies:

- For every \( u, v \in V \), \( f(u, v) \leq c(u, v) \)

Flow

How to find a Maximum Flow?

6.6: Maximum flow T.S.
A flow is a function $f : V \times V \rightarrow \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$

How to find a Maximum Flow?
A flow is a function \( f : V \times V \rightarrow \mathbb{R} \) that satisfies:

- For every \( u, v \in V \), \( f(u, v) \leq c(u, v) \)
- For every \( v \in V \setminus \{s, t\} \), \( \sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u) \)

**Flow Conservation**

The value of a flow is defined as

\[ |f| = \sum_{v \in V} f(s, v) \]

**Flow Network**

6.6: Maximum flow
A flow is a function $f : V \times V \rightarrow \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$

Flow Conservation

Flow Network

For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$

How to find a Maximum Flow?
Flow Network

A flow is a function $f : V \times V \rightarrow \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$

Flow Conservation
A flow is a function \( f : V \times V \rightarrow \mathbb{R} \) that satisfies:

- For every \( u, v \in V \), \( f(u, v) \leq c(u, v) \)
- For every \( v \in V \setminus \{s, t\} \), \( \sum_{(u, v) \in E} f(u, v) = \sum_{(v, u) \in E} f(v, u) \)

Flow Conservation

How to find a Maximum Flow?
A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

How to find a Maximum Flow?
Flow Network

Flow

A flow is a function \( f : V \times V \rightarrow \mathbb{R} \) that satisfies:

- For every \( u, v \in V \), \( f(u, v) \leq c(u, v) \)
- For every \( v \in V \setminus \{s, t\} \), \( \sum_{(u,v)\in E} f(u,v) = \sum_{(v,u)\in E} f(v,u) \)
- For every \( u, v \in V \), \( f(u, v) = -f(v, u) \)

The value of a flow is defined as \( |f| = \sum_{v\in V} f(s,v) \)
A flow is a function $f : V \times V \to \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v)\in E} f(u,v) = \sum_{(v,u)\in E} f(v,u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$.
Flow Network

A flow is a function \( f : V \times V \rightarrow \mathbb{R} \) that satisfies:

- For every \( u, v \in V \), \( f(u, v) \leq c(u, v) \)
- For every \( v \in V \setminus \{s, t\} \), \( \sum_{(u,v) \in E} f(u,v) = \sum_{(v,u) \in E} f(v,u) \)
- For every \( u, v \in V \), \( f(u, v) = -f(v, u) \)

The value of a flow is defined as \( |f| = \sum_{v \in V} f(s, v) \)
Flow Network

A flow is a function $f : V \times V \rightarrow \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$

How to find a Maximum Flow?
A flow is a function $f : V \times V \rightarrow \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u, v) \in E} f(u, v) = \sum_{(v, u) \in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$
Flow Network

A flow is a function \( f : V \times V \rightarrow \mathbb{R} \) that satisfies:

- For every \( u, v \in V \), \( f(u, v) \leq c(u, v) \)
- For every \( v \in V \setminus \{s, t\} \), \( \sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u) \)
- For every \( u, v \in V \), \( f(u, v) = -f(v, u) \)

The value of a flow is defined as \( |f| = \sum_{v \in V} f(s, v) \)

The value of the flow is \( |f| = 8 + 10 + 10 = 28 \)
A flow is a function $f : V \times V \rightarrow \mathbb{R}$ that satisfies:

- For every $u, v \in V$, $f(u, v) \leq c(u, v)$
- For every $v \in V \setminus \{s, t\}$, $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$
- For every $u, v \in V$, $f(u, v) = -f(v, u)$

The value of a flow is defined as $|f| = \sum_{v \in V} f(s, v)$

How to find a Maximum Flow?

Flow Network

6.6: Maximum flow
A First Attempt

**Greedy Algorithm**

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

| $f$ | $|f|$ |
|-----|------|
| 0   | 0    |
| 4/8 | 8    |
| 10  | 16   |
| 19  |      |

Is this optimal? Greedy did not succeed!

6.6: Maximum flow
A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

<table>
<thead>
<tr>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>19</td>
</tr>
</tbody>
</table>

Is this optimal?

Greedy did not succeed!

$|f| = 0$
A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

| $f$ | $|f|$ |
|-----|------|
| 0   | 0    |
| 8/10| 8/10 |
| 8/10| 8/10 |
| 0/10| 0/10 |
| 0/9  | 0/9  |
| 0/6  | 0/6  |
| 0/10 | 0/10 |
| 0/10 | 0/10 |
| 0/10 | 0/10 |
| 0/10 | 0/10 |
| 0/10 | 0/10 |

Greedy did not succeed!
A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

Greedy did not succeed!

<table>
<thead>
<tr>
<th>$f$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8/10</td>
<td>8/8</td>
</tr>
<tr>
<td>0/10</td>
<td>0/2</td>
</tr>
<tr>
<td>0/6</td>
<td>0/9</td>
</tr>
<tr>
<td>0/4</td>
<td>0/10</td>
</tr>
</tbody>
</table>

$|f| = 8$
A First Attempt

**Greedy Algorithm**

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

---

6.6: Maximum flow
A First Attempt

**Greedy Algorithm**

- Start with \( f(u, v) = 0 \) everywhere
- Repeat as long as possible:
  - Find a \((s, t)\)-path \(p\) where each edge \(e = (u, v)\) has \(f(u, v) < c(u, v)\)
  - Augment flow along \(p\)

---

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>0/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0/9</td>
<td>8/8</td>
<td>2/9</td>
</tr>
<tr>
<td>4</td>
<td>0/6</td>
<td>10/10</td>
<td>0/10</td>
</tr>
<tr>
<td>5</td>
<td>0/10</td>
<td>10/10</td>
<td>8/10</td>
</tr>
<tr>
<td>0/2</td>
<td></td>
<td>0/2</td>
<td></td>
</tr>
<tr>
<td>2/2</td>
<td>8/10</td>
<td>0/10</td>
<td></td>
</tr>
<tr>
<td>10/10</td>
<td></td>
<td></td>
<td>0/10</td>
</tr>
</tbody>
</table>

\(|f| = 8\)
A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

| $|f|$ |
|-----|
| 10  |

| 6.6: Maximum flow | T.S. |
A First Attempt

**Greedy Algorithm**

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

---

![Graph](image-url)

$|f| = 10$
A First Attempt

**Greedy Algorithm**

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

![Graph](image)

|$f| = 10$
A First Attempt

Greedy Algorithm

- Start with $f(u, v) = 0$ everywhere
- Repeat as long as possible:
  - Find a $(s, t)$-path $p$ where each edge $e = (u, v)$ has $f(u, v) < c(u, v)$
  - Augment flow along $p$

Is this optimal?

Greedy did not succeed!

| $f$ | 16 |
---|---|
| $s$ | 6/10 |
| 3 | 2/2 |
| 2 | 0/4 |
| 4 | 6/6 |
| 5 | 8/9 |
| t | 10/10 |
A First Attempt

Greedy Algorithm

- Start with \( f(u, v) = 0 \) everywhere
- Repeat as long as possible:
  - Find a \((s, t)\)-path \( p \) where each edge \( e = (u, v) \) has \( f(u, v) < c(u, v) \)
  - Augment flow along \( p \)

Is this optimal?

|f| = 16
A First Attempt

**Greedy Algorithm**

- Start with \( f(u, v) = 0 \) everywhere
- Repeat as long as possible:
  - Find a \((s, t)\)-path \( p \) where each edge \( e = (u, v) \) has \( f(u, v) < c(u, v) \)
  - Augment flow along \( p \)

![Graph with flow values](image)

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Flow In</th>
<th>Flow Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>t</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

\[ |f| = 19 \]

Greedy did not succeed!
Outline

Introduction

Ford-Fulkerson

Max-Flow Min-Cut Theorem
Residual Graph

Original Edge

Edge \( e = (u, v) \in E \)
- flow \( f(u, v) \) and capacity \( c(u, v) \)
Residual Graph

Original Edge

Edge $e = (u, v) \in E$
- flow $f(u, v)$ and capacity $c(u, v)$

Graph $G$:

![Graph diagram](image)
Residual Graph

Original Edge

Edge \( e = (u, v) \in E \)
- flow \( f(u, v) \) and capacity \( c(u, v) \)

Residual Capacity

\[
c_f(u, v) = \begin{cases} 
    c(u, v) - f(u, v) & \text{if } (u, v) \in E, \\
    f(v, u) & \text{if } (v, u) \in E, \\
    0 & \text{otherwise}.
\end{cases}
\]

Graph \( G \):

\[ u \xrightarrow{6/17} v \]
Residual Graph

Original Edge

Edge \( e = (u, v) \in E \)
- flow \( f(u, v) \) and capacity \( c(u, v) \)

Residual Capacity

\[
c_f(u, v) = \begin{cases} 
    c(u, v) - f(u, v) & \text{if } (u, v) \in E, \\
    f(v, u) & \text{if } (v, u) \in E, \\
    0 & \text{otherwise.}
\end{cases}
\]

Graph \( G \):

\[
\begin{array}{c}
\text{u} \\
\rightarrow \\
\text{v}
\end{array}
\]

Residual \( G_f \):

\[
\begin{array}{c}
\text{u} \\
\rightarrow \\
\text{v}
\end{array}
\]
Residual Graph

Original Edge

Edge \( e = (u, v) \in E \)
- flow \( f(u, v) \) and capacity \( c(u, v) \)

Residual Capacity

\[
c_f(u, v) = \begin{cases} 
  c(u, v) - f(u, v) & \text{if } (u, v) \in E , \\
  f(v, u) & \text{if } (v, u) \in E , \\
  0 & \text{otherwise}.
\end{cases}
\]

Residual Graph

- \( G_f = (V, E_f, c_f) \), \( E_f := \{(u, v) : c_f(u, v) > 0\} \)
Residual Graph with anti-parallel edges

Original Edge

Edge $e = (u, v) \in E$ (& possibly $e' = (v, u) \in E$)

- flow $f(u, v)$ and capacity $c(u, v)$

Graph $G$: 

![Graph diagram]

- $u$ to $v$ with flow $6/17$
- $v$ to $u$ with flow $2/4$

Residual Graph

Graph $G_f$: 

- $u$ to $v$ with residual capacity $17 - (6 - 2) = 13$
- $v$ to $u$ with residual capacity $8$

6.6: Maximum flow
Residual Graph with anti-parallel edges

Original Edge

Edge \( e = (u, v) \in E \) (\& possibly \( e' = (v, u) \in E \))
- flow \( f(u, v) \) and capacity \( c(u, v) \)

Residual Capacity

For every pair \((u, v) \in V \times V\),

\[
c_f(u, v) = c(u, v) - f(u, v).
\]
Residual Graph with anti-parallel edges

Original Edge

Edge \( e = (u, v) \in E \) (\& possibly \( e' = (v, u) \in E \))
- flow \( f(u, v) \) and capacity \( c(u, v) \)

Residual Capacity

For every pair \((u, v) \in V \times V\),

\[
c_f(u, v) = c(u, v) - f(u, v).
\]
Residual Graph with anti-parallel edges

Original Edge

Edge \( e = (u, v) \in E \) (& possibly \( e' = (v, u) \in E \))
- flow \( f(u, v) \) and capacity \( c(u, v) \)

Residual Capacity

For every pair \( (u, v) \in V \times V \),

\[
c_f(u, v) = c(u, v) - f(u, v).
\]

Graph \( G: \)

\( u \rightarrow v \)
\( 6/17 \)
\( 2/4 \)

Residual \( G_f: \)

\( u \rightarrow v \)
\( 17-(6-2) \)
\( 4-(2-6) \)
Residual Graph with anti-parallel edges

Original Edge

- Edge \( e = (u, v) \in E \) (\& possibly \( e' = (v, u) \in E \))
- flow \( f(u, v) \) and capacity \( c(u, v) \)

Residual Capacity

- For every pair \((u, v) \in V \times V\),
  \[
  c_f(u, v) = c(u, v) - f(u, v).
  \]
Residual Graph with anti-parallel edges

Original Edge

Edge $e = (u, v) \in E$ (and possibly $e' = (v, u) \in E$
- flow $f(u, v)$ and capacity $c(u, v)$

Residual Capacity

For every pair $(u, v) \in V \times V$,

$$c_f(u, v) = c(u, v) - f(u, v).$$

Residual Graph

- $G_f = (V, E_f, c_f)$, $E_f := \{(u, v) : c_f(u, v) > 0\}$

Graph $G$:

- $u \rightarrow v$ with flow $6/17$
- $u \rightarrow v$ with capacity $2/4$

Residual $G_f$:

- $u \rightarrow v$ with residual capacity $13$
- $u \rightarrow v$ with residual capacity $8$
Example of a Residual Graph (Handout)

Flow network $G$

Residual Graph $G_f$
The Ford-Fulkerson Method ("Enhanced Greedy")

0: def fordFulkerson(G)
1:   initialize flow to 0 on all edges
2:   while an augmenting path in $G_f$ can be found:
3:     push as much extra flow as possible through it
The Ford-Fulkerson Method ("Enhanced Greedy")

```python
0: def fordFulkerson(G)
1:     initialize flow to 0 on all edges
2:     while an augmenting path in $G_f$ can be found:
3:         push as much extra flow as possible through it
```

**Augmenting path:** Path from source to sink in $G_f$
The Ford-Fulkerson Method ("Enhanced Greedy")

0: def fordFulkerson(G)
1:     initialize flow to 0 on all edges
2:     while an augmenting path in $G_f$ can be found:
3:         push as much extra flow as possible through it

If $f'$ is a flow in $G_f$ and $f$ a flow in $G$, then $f + f'$ is a flow in $G$
The Ford-Fulkerson Method ("Enhanced Greedy")

0: def fordFulkerson(G)
1:     initialize flow to 0 on all edges
2:     while an augmenting path in $G_f$ can be found:
3:         push as much extra flow as possible through it

Questions:
- How to find an augmenting path?
- Does this method terminate?
- If it terminates, how good is the solution?

Using BFS or DFS, we can find an augmenting path in $O(V + E)$ time.
The Ford-Fulkerson Method ("Enhanced Greedy")

0: def fordFulkerson(G)
1:     initialize flow to 0 on all edges
2:     while an augmenting path in $G_f$ can be found:
3:         push as much extra flow as possible through it

Using BFS or DFS, we can find an augmenting path in $O(V + E)$ time.

Questions:
- How to find an augmenting path?
- Does this method terminate?
- If it terminates, how good is the solution?
Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
**Illustration of the Ford-Fulkerson Method**

**Graph** $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

**Graph** $G = (V, E, c)$:

**Residual Graph** $G_f = (V, E_f, c_f)$:

Is this a max-flow?
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

**Graph** \( G = (V, E, c) \):

\[
\begin{align*}
\text{s} & \to \text{2} & 10/10 \\
\text{2} & \to \text{4} & 0/4 \\
\text{2} & \to \text{3} & 2/2 \\
\text{3} & \to \text{4} & 8/8 \\
\text{3} & \to \text{5} & 8/9 \\
\text{5} & \to \text{t} & 10/10 \\
\text{4} & \to \text{t} & 6/6 \\
\text{5} & \to \text{t} & 6/10
\end{align*}
\]

\(|f| = 16\)

**Residual Graph** \( G_f = (V, E_f, c_f) \):

\[
\begin{align*}
\text{s} & \to \text{2} & 10 \\
\text{2} & \to \text{3} & 2 \\
\text{3} & \to \text{4} & 4 \\
\text{3} & \to \text{5} & 8 \\
\text{4} & \to \text{5} & 6 \\
\text{5} & \to \text{s} & 6 \\
\text{5} & \to \text{t} & 10 \\
\text{t} & \to \text{4} & 6 \\
\text{t} & \to \text{s} & 4 \\
\text{t} & \to \text{3} & 1 \\
\text{t} & \to \text{5} & 10
\end{align*}
\]
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:

$|f| = 16$
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$: |f| = 18
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

Residual Graph $G_f = (V, E_f, c_f)$:
Illustration of the Ford-Fulkerson Method

Graph $G = (V, E, c)$:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>t</td>
</tr>
<tr>
<td>10/10</td>
<td>3/4</td>
<td>0/2</td>
<td>9/10</td>
<td>10/10</td>
</tr>
<tr>
<td>9/10</td>
<td>7/8</td>
<td>6/6</td>
<td>9/9</td>
<td>10/10</td>
</tr>
</tbody>
</table>

$|f| = 19$

Is this a max-flow?

Residual Graph $G_f = (V, E_f, c_f)$:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>t</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>2</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

$|f| = 19$
Illustration of the Ford-Fulkerson Method

**Graph** $G = (V, E, c)$:

```
Graph G = (V, E, c):

<table>
<thead>
<tr>
<th></th>
<th>s</th>
<th>3</th>
<th>2</th>
<th>4</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>9/10</td>
<td>9/10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>10/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>3/4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9/10</td>
</tr>
<tr>
<td>t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

$|f| = 19$

Is this a max-flow?

**Residual Graph** $G_f = (V, E_f, c_f)$:

```
Residual Graph Gf = (V, Ef, cf):

<table>
<thead>
<tr>
<th></th>
<th>s</th>
<th>3</th>
<th>2</th>
<th>4</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>9</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

6.6: Maximum flow
Illustration of the Ford-Fulkerson Method

**Graph** $G = (V, E, c)$:

![Graph](image)

$|f| = 19$

Is this a max-flow?

**Residual Graph** $G_f = (V, E_f, c_f)$:

![Residual Graph](image)
Outline

Introduction

Ford-Fulkerson

Max-Flow Min-Cut Theorem
A cut \((S, T)\) is a partition of \(V\) into \(S\) and \(T = V \setminus S\) such that \(s \in S\) and \(t \in T\).

**Graph** \(G = (V, E, c)\):

- \((s, 3)\) with capacity 10
- \((3, 2)\) with capacity 2
- \((2, 4)\) with capacity 4
- \((4, 5)\) with capacity 6
- \((5, t)\) with capacity 10
- \((s, 5)\) with capacity 9
- \((2, 5)\) with capacity 8
- \((3, t)\) with capacity 10

\[c(S, T) = 10 + 9 = 19\]

\[|f| = 16\]
A cut \((S, T)\) is a partition of \(V\) into \(S\) and \(T = V \setminus S\) such that \(s \in S\) and \(t \in T\).

**Graph** \(G = (V, E, c)\):

- \(s\) to \(2\): 10
- \(2\) to \(4\): 4
- \(2\) to \(3\): 2
- \(2\) to \(t\): 8
- \(3\) to \(s\): 10
- \(3\) to \(5\): 9
- \(4\) to \(5\): 6
- \(4\) to \(t\): 10
- \(5\) to \(t\): 10
From Flows to Cuts

Cut

- A cut \((S, T)\) is a partition of \(V\) into \(S\) and \(T = V \setminus S\) such that \(s \in S\) and \(t \in T\).
- The capacity of a cut \((S, T)\) is the sum of capacities of the edges from \(S\) to \(T\):

\[
c(S, T) = \sum_{u \in S, v \in T} c(u, v) = \sum_{(u,v) \in E(S,T)} c(u, v)
\]

Graph \(G = (V, E, c)\):

\[
c(\{s, 3\}, \{2, 4, 5, t\}) = \]

6.6: Maximum flow

T.S.
A cut \((S, T)\) is a partition of \(V\) into \(S\) and \(T = V \setminus S\) such that \(s \in S\) and \(t \in T\).

The capacity of a cut \((S, T)\) is the sum of capacities of the edges from \(S\) to \(T\):

\[
c(S, T) = \sum_{u \in S, v \in T} c(u, v) = \sum_{(u,v) \in E(S,T)} c(u, v)
\]

**Graph** \(G = (V, E, c)\):

\[
c(\{s, 3\}, \{2, 4, 5, t\}) = 10 + 9 = 19
\]
A cut \((S, T)\) is a partition of \(V\) into \(S\) and \(T = V \setminus S\) such that \(s \in S\) and \(t \in T\).

The capacity of a cut \((S, T)\) is the sum of capacities of the edges from \(S\) to \(T\):

\[
c(S, T) = \sum_{u \in S, v \in T} c(u, v) = \sum_{(u, v) \in E(S, T)} c(u, v)
\]

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

Graph \(G = (V, E, c)\):

\[
|f| = 16
\]
From Flows to Cuts

**Flow Value Lemma (Lemma 26.4)**

Let $f$ be a flow with source $s$ and sink $t$, and let $(S, T)$ be any cut of $G$. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v) \in E(S,T)} f(u,v) - \sum_{(v,u) \in E(T,S)} f(v,u).$$

---

**Graph** $G = (V, E, c)$:

$|f| = 16$
Flow Value Lemma (Lemma 26.4)

Let \( f \) be a flow with source \( s \) and sink \( t \), and let \((S, T)\) be any cut of \( G \). Then the value of the flow is equal to the net flow across the cut, i.e.,

\[
|f| = \sum_{(u, v) \in E(S, T)} f(u, v) - \sum_{(v, u) \in E(T, S)} f(v, u).
\]

Graph \( G = (V, E, c) \):

| \( f \) = 16 |
Flow Value Lemma (Lemma 26.4)

Let $f$ be a flow with source $s$ and sink $t$, and let $(S, T)$ be any cut of $G$. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u).$$

Graph $G = (V, E, c)$:

$$|f| = 16$$
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let $f$ be a flow with source $s$ and sink $t$, and let $(S, T)$ be any cut of $G$. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v) \in E(S,T)} f(u,v) - \sum_{(v,u) \in E(T,S)} f(v,u).$$

Graph $G = (V, E, c)$:

$$|f| = 16$$

10 - 2 + 8 = 16
Flow Value Lemma (Lemma 26.4)

Let $f$ be a flow with source $s$ and sink $t$, and let $(S, T)$ be any cut of $G$. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u, v) \in E(S, T)} f(u, v) - \sum_{(v, u) \in E(T, S)} f(v, u).$$

Graph $G = (V, E, c)$:

$$|f| = 16$$
From Flows to Cuts

Flow Value Lemma (Lemma 26.4)

Let $f$ be a flow with source $s$ and sink $t$, and let $(S, T)$ be any cut of $G$. Then the value of the flow is equal to the net flow across the cut, i.e.,

$$|f| = \sum_{(u,v) \in E(S,T)} f(u,v) - \sum_{(v,u) \in E(T,S)} f(v,u).$$

Graph $G = (V, E, c)$:

$$|f| = 16$$
Flow Value Lemma (Lemma 26.4)

Let \( f \) be a flow with source \( s \) and sink \( t \), and let \((S, T)\) be any cut of \( G \). Then the value of the flow is equal to the net flow across the cut, i.e.,

\[
|f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u).
\]

Graph \( G = (V, E, c) \):

\[
8 + 8 - 6 + 6 = 16
\]
From Flows to Cuts

|\begin{align*}
|f| &= \sum_{(u,v) \in E(S,T)} f(u,v) - \sum_{(v,u) \in E(T,S)} f(v,u).
|\end{align*}|

Graph $G = (V, E, c)$:

\[8 + 8 - 6 + 6 = 16\]
From Flows to Cuts

\[ |f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u). \]

Graph \( G = (V, E, c) \):

\[ |f| = 16 \]

\[ 8 + 8 - 6 + 6 = 16 \]
From Flows to Cuts

\[ |f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u). \]

\[ |f| = \sum_{w \in V} f(s, w) \]

**Graph** \( G = (V, E, c) \):

\[ |f| = 16 \]

\[ 8 + 8 - 6 + 6 = 16 \]
From Flows to Cuts

\[ |f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u). \]

\[ |f| = \sum_{w \in V} f(s, w) = \sum_{u \in S} \left( \sum_{(u,w) \in E} f(u, w) - \sum_{(w,u) \in E} f(w, u) \right) \]

**Graph** \( G = (V, E, c) \):

\[ |f| = 16 \]

\[ 8 + 8 - 6 + 6 = 16 \]
From Flows to Cuts

\[ |f| = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u). \]

\[ |f| = \sum_{w \in V} f(s, w) = \sum_{u \in S} \left( \sum_{(u,w) \in E} f(u, w) - \sum_{(w,u) \in E} f(w, u) \right) \]

\[ = \sum_{(u,v) \in E(S,T)} f(u, v) - \sum_{(v,u) \in E(T,S)} f(v, u). \]

**Graph** \( G = (V, E, c) : \)

\[ |f| = 16 \]

\[ 8 + 8 - 6 + 6 = 16 \]