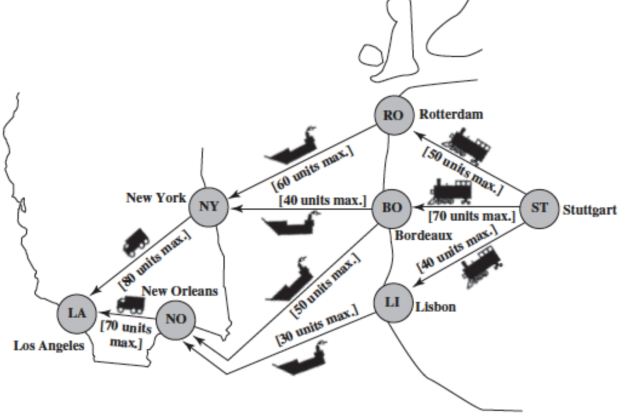
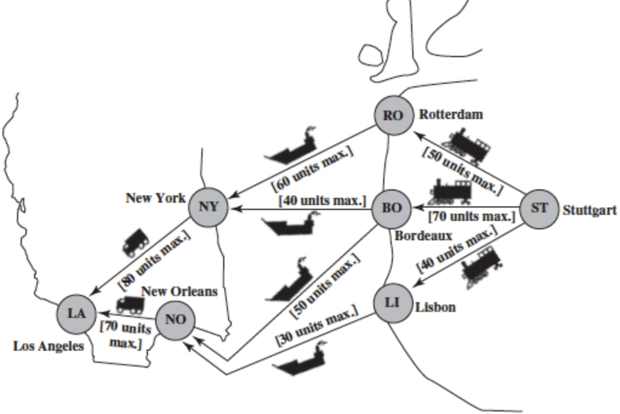
Algorithms and Complexity

Maximum Flow and Flow Networks

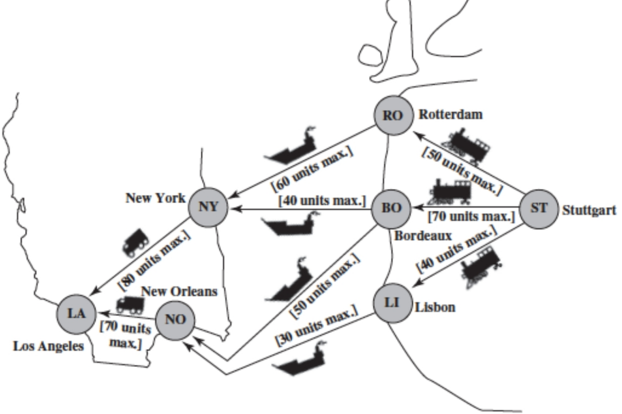
Marc Hellmuth

University of Stockholm

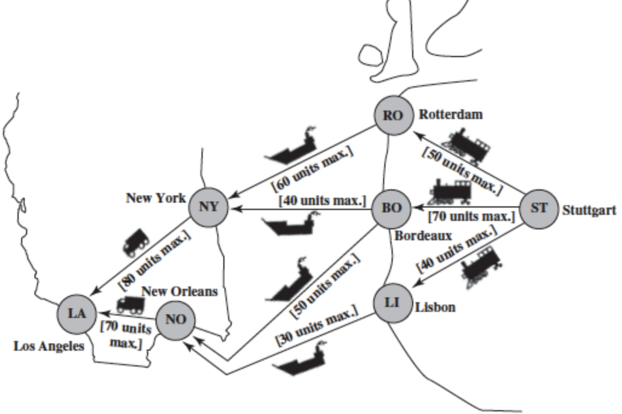

RO
Rotterdam
[50 units max.]
[60 units max.]
New York
NY
[40 units max.]
BO
[70 units max.]
ST
Stuttgart
Bordeaux
[40 units max.]
[80 units max.]
New Orleans
[50 units max.]
LI
Lisbon
LA
NO
[70 units max.]
[30 units max.]
Los Angeles



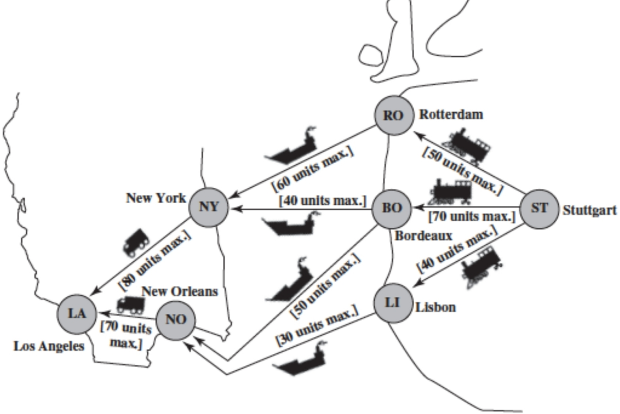
- a material (e.g. a liquid, electrical current, countable items) is produced at a source



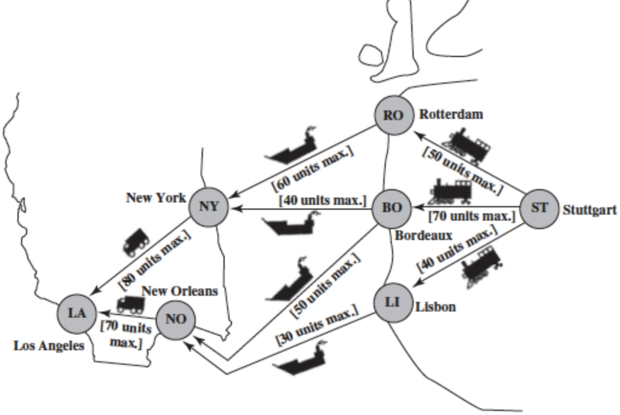
- a material (e.g. a liquid, electrical current, countable items) is produced at a source
- it is moving through a system (e.g. of pipes, wires, roads) that is modeled as a directed graph



- a material (e.g. a liquid, electrical current, countable items) is produced at a source
- it is moving through a system (e.g. of pipes, wires, roads) that is modeled as a directed graph
- each edge has a given capacity (e.g. diameter of pipe, wire, capacity of road) that limits the amount of material that can pass this edge



- a material (e.g. a liquid, electrical current, countable items) is produced at a source
- it is moving through a system (e.g. of pipes, wires, roads) that is modeled as a directed graph
- each edge has a given capacity (e.g. diameter of pipe, wire, capacity of road) that limits the amount of material that can pass this edge
- the system is in a steady state, i.e. the amount of material leaving a vertex is equal to the amount of material entering the vertex [= no possibility to store material]



- a material (e.g. a liquid, electrical current, countable items) is produced at a source
- it is moving through a system (e.g. of pipes, wires, roads) that is modeled as a directed graph
- each edge has a given capacity (e.g. diameter of pipe, wire, capacity of road) that limits the amount of material that can pass this edge
- the system is in a steady state, i.e. the amount of material leaving a vertex is equal to the amount of material entering the vertex [= no possibility to store material]
- all material ends at a vertex called target, where it is consumed

Flow Network and Maximum Flow

A flow network $G = (V, E)$ is a directed graph such that

- G has no self-loops
- $(u, v) \in E \Rightarrow (v, u) \notin E \qquad (\forall u, v \in V)$
- V has two distinct vertices, a source s and a target t
- $\forall v \in V$ there is a path $s \rightsquigarrow v \rightsquigarrow t$

Flow Network and Maximum Flow

A flow network $G = (V, E)$ is a directed graph such that

- G has no self-loops
- $(u, v) \in E \Rightarrow (v, u) \notin E \qquad (\forall u, v \in V)$
- V has two distinct vertices, a source s and a target t
- $\forall v \in V$ there is a path $s \rightsquigarrow v \rightsquigarrow t$

For a di-graph $G = (V, E)$ a capacity funktion is a map $c\colon V \times V \to \mathbb{R}$ such that

- each edge $(u, v) \in E$ has a capacity $c(u, v) \geq 0$
- we define $c(u, v) := 0$ if $(u, v) \notin E$

Flow Network and Maximum Flow

A flow network $G = (V,E)$ is a directed graph such that

- G has no self-loops
- $(u,v) \in E \Rightarrow (v,u) \notin E \qquad (\forall u,v \in V)$
- V has two distinct vertices, a source s and a target t
- $\forall v \in V$ there is a path $s \leadsto v \leadsto t$

For a di-graph $G = (V,E)$ a capacity funktion is a map $c\colon V \times V \to \mathbb{R}$ such that

- each edge $(u,v) \in E$ has a capacity $c(u,v) \geq 0$
- we define $c(u,v) := 0$ if $(u,v) \notin E$

For a di-graph $G = (V,E)$ with cap.fct. c, a flow in G is a function $f : V \times V \to \mathbb{R}$ such that

F1 capacity constraint: $0 \leq f(u,v) \leq c(u,v) \qquad (\forall u,v \in V)$
F2 flow conservation: $\sum_{v \in V} f(u,v) = \sum_{v \in V} f(v,u) \qquad (u \in V \setminus \{s,t\})$

Flow Network and Maximum Flow

A flow network $G = (V, E)$ is a directed graph such that

- G has no self-loops
- $(u, v) \in E \Rightarrow (v, u) \notin E \qquad (\forall u, v \in V)$
- V has two distinct vertices, a source s and a target t
- $\forall v \in V$ there is a path $s \rightsquigarrow v \rightsquigarrow t$

For a di-graph $G = (V, E)$ a capacity funktion is a map $c\colon V \times V \to \mathbb{R}$ such that

- each edge $(u, v) \in E$ has a capacity $c(u, v) \geq 0$
- we define $c(u, v) := 0$ if $(u, v) \notin E$

For a di-graph $G = (V, E)$ with cap.fct. c, a flow in G is a function $f : V \times V \to \mathbb{R}$ such that

F1 capacity constraint: $0 \leq f(u, v) \leq c(u, v) \qquad (\forall u, v \in V)$

F2 flow conservation: $\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u) \qquad (u \in V \setminus \{s, t\})$

Obs. 1: $(u, v) \notin E \implies c(u, v) = 0 \implies f(u, v) = 0$

Flow Network and Maximum Flow

A flow network $G = (V, E)$ is a directed graph such that

- G has no self-loops
- $(u, v) \in E \Rightarrow (v, u) \notin E \qquad (\forall u, v \in V)$
- V has two distinct vertices, a source s and a target t
- $\forall v \in V$ there is a path $s \rightsquigarrow v \rightsquigarrow t$

For a di-graph $G = (V, E)$ a capacity funktion is a map $c\colon V \times V \to \mathbb{R}$ such that

- each edge $(u, v) \in E$ has a capacity $c(u, v) \geq 0$
- we define $c(u, v) := 0$ if $(u, v) \notin E$

For a di-graph $G = (V, E)$ with cap.fct. c, a flow in G is a function $f : V \times V \to \mathbb{R}$ such that

F1 capacity constraint: $0 \leq f(u, v) \leq c(u, v) \qquad (\forall u, v \in V)$

F2 flow conservation: $\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u) \qquad (u \in V \setminus \{s, t\})$

Obs. 1: $(u, v) \notin E \implies c(u, v) = 0 \implies f(u, v) = 0$

Aim is to maximize flow, i.e., maximize

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

Flow Network and Maximum Flow

A flow network $G = (V, E)$ is a directed graph such that

- G has no self-loops
- $(u, v) \in E \Rightarrow (v, u) \notin E \qquad (\forall u, v \in V)$
- V has two distinct vertices, a source s and a target t
- $\forall v \in V$ there is a path $s \leadsto v \leadsto t$

For a di-graph $G = (V, E)$ a capacity funktion is a map $c \colon V \times V \to \mathbb{R}$ such that

- each edge $(u, v) \in E$ has a capacity $c(u, v) \geq 0$
- we define $c(u, v) := 0$ if $(u, v) \notin E$

For a di-graph $G = (V, E)$ with cap.fct. c, a flow in G is a function $f : V \times V \to \mathbb{R}$ such that

F1 capacity constraint: $0 \leq f(u, v) \leq c(u, v) \qquad (\forall u, v \in V)$

F2 flow conservation: $\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u) \qquad (u \in V \setminus \{s, t\})$

Obs. 1: $(u, v) \notin E \implies c(u, v) = 0 \implies f(u, v) = 0$

Aim is to maximize flow, i.e., maximize

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

In practice, we often have multiple sourses, targets or anti-parallel edges – no problem!
[WHITEBOARD]

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FORDFULKERSON_METHOD(G, s, t, c)

Init $f(u,v) = 0$ for all $(u,v) \in E$
while $\exists$ augmenting path P in residual network G_f **do**
    Augment flow f along P
return f

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FordFulkerson_Method(G, s, t, c)

Init $f(u,v) = 0$ for all $(u,v) \in E$
while $\exists$ augmenting path P in residual network G_f **do**
  Augment flow f along P
return f

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FordFulkerson_Method(G, s, t, c)

Init $f(u, v) = 0$ for all $(u, v) \in E$
while $\exists$ augmenting path P in residual network G_f **do**
    Augment flow f along P
return f

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FORDFULKERSON_METHOD(G, s, t, c)

```
Init f(u,v) = 0 for all (u,v) ∈ E
while ∃ augmenting path P in residual network G_f do
    Augment flow f along P
return f
```

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FORDFULKERSON_METHOD(G, s, t, c)

```
Init f(u,v) = 0 for all (u,v) ∈ E
while ∃ augmenting path P in residual network G_f do
    Augment flow f along P
return f
```

Let f be a flow in G. Then for two vertices u, v the residual capacity is defined as

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

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FORDFULKERSON_METHOD(G, s, t, c)

1: Init $f(u,v) = 0$ for all $(u,v) \in E$
2: **while** $\exists$ augmenting path P in residual network G_f **do**
3: Augment flow f along P
4: **return** f

Let f be a flow in G. Then for two vertices u, v the residual capacity is defined as

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

The residual network of G induced by f is $G_f = (V, E_f)$, where

$$E_f = \{(u,v) \in V \times V \,|\, c_f(u,v) > 0\}.$$

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FORDFULKERSON_METHOD(G, s, t, c)

```
Init f(u,v) = 0 for all (u,v) ∈ E
while ∃ augmenting path P in residual network G_f do
    Augment flow f along P
return f
```

Let f be a flow in G. Then for two vertices u, v the residual capacity is defined as

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

The residual network of G induced by f is $G_f = (V, E_f)$, where

$$E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}.$$

Intuition: G_f consists of all edges of G where we can increase flow and edges (non-edges of G) to represent the possibilty to decrease flow, i.e., an edge to admit positive flow in opposite direction

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c.

FORDFULKERSON_METHOD(G, s, t, c)

Init $f(u,v) = 0$ for all $(u,v) \in E$
while $\exists$ augmenting path P in residual network G_f **do**
    Augment flow f along P
return f

Let f be a flow in G. Then for two vertices u, v the residual capacity is defined as

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

The residual network of G induced by f is $G_f = (V, E_f)$, where

$$E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}.$$

Intuition: G_f consists of all edges of G where we can increase flow and edges (non-edges of G) to represent the possibilty to decrease flow, i.e., an edge to admit positive flow in opposite direction

Obs. 2: $|E_f| \leq 2|E|$

Ford-Fulkerson Method and Residual Network

Given flow network $G = (V,E)$ with source s, target t and capacity fct. c.

FORDFULKERSON_METHOD(G, s, t, c)

1: Init $f(u,v) = 0$ for all $(u,v) \in E$
2: **while** $\exists$ augmenting path P in residual network G_f **do**
3: Augment flow f along P
4: **return** f

Let f be a flow in G. Then for two vertices u, v the residual capacity is defined as

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

The residual network of G induced by f is $G_f = (V, E_f)$, where

$$E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}.$$

Intuition: G_f consists of all edges of G where we can increase flow and edges (non-edges of G) to represent the possibilty to decrease flow, i.e., an edge to admit positive flow in opposite direction

Obs. 2: $|E_f| \leq 2|E|$

Flows in the Residual Network and Augmentation

A flow f' in the residual network G_f can be used to add flow to flow f in the original flow network G.

Flows in the Residual Network and Augmentation

A flow f' in the residual network G_f can be used to add flow to flow f in the original flow network G.

Let f be a flow in G and f' be a flow in the residual network G_f. Define $f \uparrow f' : V \times V \to \mathbb{R}$ – the augmentation of f by f' – as follows

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

Flows in the Residual Network and Augmentation

A flow f' in the residual network G_f can be used to add flow to flow f in the original flow network G.

Let f be a flow in G and f' be a flow in the residual network G_f. Define $f \uparrow f' : V \times V \to \mathbb{R}$ – the augmentation of f by f' – as follows

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

Intuition: Increase flow $f(u,v)$ by $f'(u,v)$ and decrease by $f'(v,u)$

Flows in the Residual Network and Augmentation

A flow f' in the residual network G_f can be used to add flow to flow f in the original flow network G.

Let f be a flow in G and f' be a flow in the residual network G_f. Define $f \uparrow f' : V \times V \to \mathbb{R}$ – the augmentation of f by f' – as follows

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

Intuition: Increase flow $f(u,v)$ by $f'(u,v)$ and decrease by $f'(v,u)$

Lemma 1: Given flow network $G = (V,E)$ with source s, target t, capacity fct. c and flow f and f' be flow in G_f. Then, $f \uparrow f'$ is a flow in G and satisfies $|f \uparrow f'| = |f| + |f'|$.

[Proof - WHITEBOARD]

Flows in the Residual Network and Augmentation

Flows in the Residual Network and Augmentation

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c and flow f.

- An augmenting path P is a simple st-path in G_f.
- The residual capacity of P is

$$c_f(P) = \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}.$$

Flows in the Residual Network and Augmentation

Given flow network $G=(V,E)$ with source s, target t and capacity fct. c and flow f.

- An augmenting path P is a simple st-path in G_f.
- The residual capacity of P is

$$c_f(P) = \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}.$$

Obs. 3: (u,v) is an edge in $G_f \implies c_f(u,v) > 0 \implies c_f(P) > 0$

Flows in the Residual Network and Augmentation

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c and flow f.

- An augmenting path P is a simple st-path in G_f.
- The residual capacity of P is

$$c_f(P) = \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}.$$

Obs. 3: (u,v) is an edge in $G_f \implies c_f(u,v) > 0 \implies c_f(P) > 0$

Obs. 4: Maximum value for which we can increase flow along P in G_f is $c_f(P)$.

Flows in the Residual Network and Augmentation

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c and flow f.

- An augmenting path P is a simple st-path in G_f.
- The residual capacity of P is

$$c_f(P) = \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}.$$

Obs. 3: (u,v) is an edge in $G_f \implies c_f(u,v) > 0 \implies c_f(P) > 0$

Obs. 4: Maximum value for which we can increase flow along P in G_f is $c_f(P)$.

- The flow $f_p \colon V \times V \to \mathbb{R}$ defined by augmenting path P is

$$f_P(u,v) = \begin{cases} c_f(P) & \text{, if } (u,v) \text{ is on } P \\ 0 & \text{, otherwise.} \end{cases}$$

Flows in the Residual Network and Augmentation

Given flow network $G = (V,E)$ with source s, target t and capacity fct. c and flow f.

- An augmenting path P is a simple st-path in G_f.
- The residual capacity of P is

$$c_f(P) = \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}.$$

Obs. 3: (u,v) is an edge in $G_f \implies c_f(u,v) > 0 \implies c_f(P) > 0$

Obs. 4: Maximum value for which we can increase flow along P in G_f is $c_f(P)$.

- The flow $f_p : V \times V \to \mathbb{R}$ defined by augmenting path P is

$$f_P(u,v) = \begin{cases} c_f(P) & \text{, if } (u,v) \text{ is on } P \\ 0 & \text{, otherwise.} \end{cases}$$

Lemma 2: Given flow network $G = (V,E)$ with source s, target t, capacity fct. c, flow f and P be an augmenting path in G_f. Then, f_P is a flow in G_f with value $|f_P| = c_f(P) > 0$.

[Proof - WHITEBOARD]

Flows in the Residual Network and Augmentation

Given flow network $G = (V, E)$ with source s, target t and capacity fct. c and flow f.

- An augmenting path P is a simple st-path in G_f.
- The residual capacity of P is

$$c_f(P) = \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}.$$

Obs. 3: (u,v) is an edge in $G_f \implies c_f(u,v) > 0 \implies c_f(P) > 0$

Obs. 4: Maximum value for which we can increase flow along P in G_f is $c_f(P)$.

- The flow $f_p : V \times V \to \mathbb{R}$ defined by augmenting path P is

$$f_P(u,v) = \begin{cases} c_f(P) & \text{, if } (u,v) \text{ is on } P \\ 0 & \text{, otherwise.} \end{cases}$$

Lemma 2: Given flow network $G = (V, E)$ with source s, target t, capacity fct. c, flow f and P be an augmenting path in G_f. Then, f_P is a flow in G_f with value $|f_P| = c_f(P) > 0$.

[Proof - WHITEBOARD]

Cor. 1: Given flow network $G = (V, E)$ with source s, target t, capacity fct. c, flow f and P be an augmenting path in G_f. Then, $|f \uparrow f_P| = |f| + |f_P| > |f|$.

To recall ..

FORDFULKERSON_METHOD(G, s, t, c)

```
Init f(u,v) = 0 for all (u,v) ∈ E
while ∃ augmenting path P in residual network G_f do
    Augment flow f along P
return f
```

To recall ..

FORDFULKERSON_METHOD(G, s, t, c)

```
Init f(u,v) = 0 for all (u,v) ∈ E
while ∃ augmenting path P in residual network G_f do
    Augment flow f along P
return f
```

- Does it terminate?
- If it terminates, do we get a maximum flow?

To recall ..

FORDFULKERSON_METHOD(G, s, t, c)

```
Init f(u,v) = 0 for all (u,v) ∈ E
while ∃ augmenting path P in residual network G_f do
    Augment flow f along P
return f
```

- Does it terminate?
- If it terminates, do we get a maximum flow?

Show, that if it terminates, then we a mximum flow f of G is returned.
To this end, we need the definition of CUTS!

Cuts of Flow Networks

Given flow network $G = (V, E)$ with source s, target t, capacity fct. c and flow f.

Cuts of Flow Networks

Given flow network $G = (V, E)$ with source s, target t, capacity fct. c and flow f.

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

Cuts of Flow Networks

Given flow network $G = (V, E)$ with source s, target t, capacity fct. c and flow f.

- A cut (S, T) of G is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The net flow $f(S, T)$ across the cut (S, T) is

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

Cuts of Flow Networks

Given flow network $G = (V, E)$ with source s, target t, capacity fct. c and flow f.

- A cut (S, T) of G is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The net flow $f(S, T)$ across the cut (S, T) is

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

- The capacity $c(S, T)$ of the cut (S, T) is

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v).$$

Cuts of Flow Networks

Given flow network $G=(V,E)$ with source s, target t, capacity fct. c and flow f.

- A cut (S,T) of G is a partition of V into S and $T=V\setminus S$ such that $s\in S$ and $t\in T$.
- The net flow $f(S,T)$ across the cut (S,T) is

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

- The capacity $c(S,T)$ of the cut (S,T) is

$$c(S,T)=\sum_{u\in S}\sum_{v\in T}c(u,v).$$

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

Cuts of Flow Networks

Given flow network $G = (V,E)$ with source s, target t, capacity fct. c and flow f.

- A cut (S,T) of G is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The net flow $f(S,T)$ across the cut (S,T) is

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

- The capacity $c(S,T)$ of the cut (S,T) is

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v).$$

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

Lemma 3: Given flow network G with source s, target t, capacity fct. c and flow f.
Then, $|f| = f(S,T)$ for any cut (S,T) of G.

[Proof - WHITEBOARD]

Cuts of Flow Networks

Given flow network $G = (V, E)$ with source s, target t, capacity fct. c and flow f.

- A cut (S, T) of G is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The net flow $f(S, T)$ across the cut (S, T) is

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

- The capacity $c(S, T)$ of the cut (S, T) is

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v).$$

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

Lemma 3: Given flow network G with source s, target t, capacity fct. c and flow f.
Then, $|f| = f(S, T)$ for any cut (S, T) of G.

[Proof - WHITEBOARD]

Cor. 2: $|f| \leq c(S, T)$ for any cut (S, T) of G.

Cuts of Flow Networks

Given flow network $G = (V, E)$ with source s, target t, capacity fct. c and flow f.

- A cut (S, T) of G is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.
- The net flow $f(S, T)$ across the cut (S, T) is

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

- The capacity $c(S, T)$ of the cut (S, T) is

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v).$$

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

Lemma 3: Given flow network G with source s, target t, capacity fct. c and flow f.
Then, $|f| = f(S, T)$ for any cut (S, T) of G.

[Proof - WHITEBOARD]

Cor. 2: $|f| \leq c(S, T)$ for any cut (S, T) of G.

Thm [Max-Flow Min-Cut Theorem] Given flow network G with source s, target t, capacity fct. c and flow f. The following statements are quivalent:

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 - WHITEBOARD]

FordFulkerson_Algorithm(G, s, t, c)

```
for each edge (u,v) ∈ E do f(u,v) ← 0
while there is a path P from s to t in the residual network G_f do
    c_f(P) ← min{c_f(u,v) | (u,v) is on P}
    for each edge (u,v) in p do
        if (u,v) ∈ E then
            f(u,v) ← f(u,v) + c_f(P)
        else
            f(v,u) ← f(v,u) − c_f(P)
return f as maximum flow
```

FordFulkerson_Algorithm(G, s, t, c)

for each edge $(u,v) \in E$ **do** $f(u,v) \leftarrow 0$
while there is a path P from s to t in the residual network G_f **do**
    $c_f(P) \leftarrow \min\{c_f(u,v) \mid (u,v)$ is on $P\}$
    for each edge (u,v) in p **do**
        if $(u,v) \in E$ **then**
            $f(u,v) \leftarrow f(u,v) + c_f(P)$
        else
            $f(v,u) \leftarrow f(v,u) - c_f(P)$
return f as maximum flow

By the latter arguments, it is ensured, that if this alg. terminates, we get maximum flow.

FORDFULKERSON_ALGORITHM(G, s, t, c)

for each edge $(u,v) \in E$ **do** $f(u,v) \leftarrow 0$

while there is a path P from s to t in the residual network G_f **do**

    $c_f(P) \leftarrow \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}$

    for each edge (u,v) in p **do**

        if $(u,v) \in E$ **then**

            $f(u,v) \leftarrow f(u,v) + c_f(P)$

        else

            $f(v,u) \leftarrow f(v,u) - c_f(P)$

return f as maximum flow

By the latter arguments, it is ensured, that if this alg. terminates, we get maximum flow.

However, if we have e.g. irrational capacities, this alg. may no converge.

FordFulkerson_Algorithm(G, s, t, c)

for each edge $(u,v) \in E$ **do** $f(u,v) \leftarrow 0$
while there is a path P from s to t in the residual network G_f **do**
    $c_f(P) \leftarrow \min\{c_f(u,v) \mid (u,v) \text{ is on } P\}$
    for each edge (u,v) in p **do**
        if $(u,v) \in E$ **then**
            $f(u,v) \leftarrow f(u,v) + c_f(P)$
        else
            $f(v,u) \leftarrow f(v,u) - c_f(P)$
return f as maximum flow

By the latter arguments, it is ensured, that if this alg. terminates, we get maximum flow.

However, if we have e.g. irrational capacities, this alg. may no converge.

If integer capacities are used (or real number that are rescaled to integers), the algorithm terminates as in each step, $f \leftarrow f \uparrow f_P$ is increased by at least one.

FORDFULKERSON_ALGORITHM(G, s, t, c)

1: **for** each edge $(u,v) \in E$ **do** $f(u,v) \leftarrow 0$
2: **while** there is a path P from s to t in the residual network G_f **do**
3: $\quad c_f(P) \leftarrow \min\{c_f(u,v) \mid (u,v)$ is on $P\}$
4: $\quad$ **for** each edge (u,v) in p **do**
5: $\quad\quad$ **if** $(u,v) \in E$ **then**
6: $\quad\quad\quad f(u,v) \leftarrow f(u,v) + c_f(P)$
7: $\quad\quad$ **else**
8: $\quad\quad\quad f(v,u) \leftarrow f(v,u) - c_f(P)$
9: return f as maximum flow

By the latter arguments, it is ensured, that if this alg. terminates, we get maximum flow.

However, if we have e.g. irrational capacities, this alg. may no converge.

If integer capacities are used (or real number that are rescaled to integers), the algorithm terminates as in each step, $f \leftarrow f \uparrow f_P$ is increased by at least one.

Runtime (assuming that f^* is returned max flow):

- Line 1: $O(|E|)$ (via BFS in $O(|V|+|E|)$ time and since G "connected" and so $|E| \geq |V|-1$")
- Line 2: is called at most $|f^*|$ times and finding st-path can be done in $O(|E|)$ time.
- Line 3: When constructing P in Line 2, we can keep track of $c_f(P)$, i.e., constant time in Line 3
- Line 4: $O(|E|)$ calls (with Line 5,6 constant time)

In total $O(|f^*||E|)$ time - which can cause problem if e.g. $|f^*| = 10^9$.

FORDFULKERSON_ALGORITHM(G, s, t, c)

```
1: for each edge (u,v) ∈ E do f(u,v) ← 0
2: while there is a path P from s to t in the residual network G_f do
3:     c_f(P) ← min{c_f(u,v) | (u,v) is on P}
4:     for each edge (u,v) in p do
5:         if (u,v) ∈ E then
6:             f(u,v) ← f(u,v) + c_f(P)
7:         else
8:             f(v,u) ← f(v,u) − c_f(P)
9: return f as maximum flow
```

By the latter arguments, it is ensured, that if this alg. terminates, we get maximum flow.

However, if we have e.g. irrational capacities, this alg. may no converge.

If integer capacities are used (or real number that are rescaled to integers), the algorithm terminates as in each step, $f \leftarrow f \uparrow f_P$ is increased by at least one.

Runtime (assuming that f^* is returned max flow):

- Line 1: $O(|E|)$ (via BFS in $O(|V|+|E|)$ time and since G ”connected” and so $|E| \geq |V|-1$”)
- Line 2: is called at most $|f^*|$ times and finding st-path can be done in $O(|E|)$ time.
- Line 3: When constructing P in Line 2, we can keep track of $c_f(P)$, i.e., constant time in Line 3
- Line 4: $O(|E|)$ calls (with Line 5,6 constant time)

In total $O(|f^*||E|)$ time - which can cause problem if e.g. $|f^*| = 10^9$.

If one chooses ”shortest st-paths” in Line 2 we obtain the EDMONDS-KARP algorithm which has runtime $O(|V||E|^2)$ time [Detail Chp 26 in Course Book]