The Max Flow Prob

Input: Flow network \( G(V, E) \) directed (oriented)

1) Edge capacities \( C : V \times V \rightarrow \mathbb{R} \)
   \( C(u, v) \geq 0 \)
   \( \forall (u, v) \notin E \) then \( C(u, v) = 0 \)

2) \( s \neq v \) \( s \equiv \text{source} & t \equiv \text{sink} \)

Goal: A flow \( f : V \times V \rightarrow \mathbb{R} \) s.t.

1) Capacity constraints \( f(u, v) \leq C(u, v) \)

2) Skewed symmetric \( f(u, v) = -f(v, u) \)

3) Flow at a vertex \( u \in V - \{s, t\} \)

\[ \sum_{v \in V} f(u, v) = 0 \text{ ie flow-in = flow-out} \]

Net flow \( = |f| = \sum_{v \in V} f(s, v) \)
Max-Flow Prob:

\text{input: } \text{Flow-Network } G = (V, E), s, t, c

\text{output: } \text{Flow } f \text{ with max net-flow}

Do examples at end of lecture.
**Residual Networks**

Network $G = (V, E)$ & flow $f$

Residual Capacity: $C_f(u,v) = C(u,v) - f(u,v)$

Residual Network:
$E_f = \{(u,v) \in V^2 \mid C_f(u,v) > 0\}$

$G_f(V, E_f)$
Ford-Fulkerson

FF Method \((G, s, t, c)\)

1) Initializing flow \(f\) to 0

2) While there is a \(p\) in \(G_f\)
   Set \(f = p + f\)

3) return \(f\)
Lemma \( f \) flow on \( G \), \( G_f \) the residual
a) \( f' \) inflow on \( G_f \) iff \( f + f' \) flow on \( G \).
b) \( f' \) is max-flow on \( G_f \) iff \( f + f' \) is max flow on \( G \).
c) \( |f + f'| = |f| + |f'| \) (\( f' \) means flow into \( S \)).

Proof
a) \( f'(e) \leq \delta_f(e) \) iff \( f'(e) \leq c(e) - f(e) \)
   \( \iff (f' + f)e \leq c(e) \)
Let $S, T$ be a cut if

1. $S \cap T = \emptyset$ and $S \cup T = V$
2. $s \in S$ and $t \in T$

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

$f$ is a flow

\[
f(x, y) = \sum_{u \in x} \sum_{v \in y} f(u, v)
\]

$f(S, T)$ is net flow from $S$ to $T$

$|f| = f(s, V-S)$
Lemma. If flow on $G$ & $(S, T)$ in a cut, then $|f| = f(S, T)$.

Proof by induction on $|S|

|S| = 1 done

Assume true for $cut(S', T')$ where $|S'| + 1 = |S|

$S = S'' \cup \{x\}$  $x \notin S'$  $x \in S$

$T' = T' - \{x\}$

$T = T \cup \{x\}$

\[
f(S, T) = f(S', T) + f(x, T)
\]

\[
f(S', T') = f(S', T) + f(S', x)
\]

\[
f(S, T) - f(S', T') = f(x, T) - f(S', x) = f(x, T) + f(x, S')
\]

\[
= f(x, T - \{x\}) = 0
\]

\[
\text{induct}
\]

\[
|f| = f(S', T') = f(S, T)
\]
Cor S, T cut \( \text{if} f \leq C(S, T) \)

since \( f(S, T) \leq C(S, T) \)

Proof

\[ f(S, T) = \text{cap}(S, T) \]

Thm max-flow = min-cut

ie the following are equivalent

1) \( f \) is max-flow
2) \( G_f \) contains no augmenting paths.
3) \( \exists \) cut \( S, T \) s.t. \( |f| = \text{cap}(S, T) \)

\[ 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 1 \]
1) \implies 2) \cup 7) \implies 7)

\exists \text{ augmenting path } \implies f \text{ is not maximum}

2) \implies 3)

Let \( S \subseteq V = \text{reachable vertices from } s \) in \( G_f \)

\( T = V - S \) note: \( t \in T \) by 2).

In \( G \)

\[ |f| = f(S, T) = \text{cap}(S, T) \]

3) \implies 1) clear
An Image Segmentation Prob

Foreground / Background Prob

Input: 1) Pixel Image
2) Affinity graph, \( G=(V,E) \)

Weights \( p_{ij} = \text{similarity of pixel } V_i \& V_j \)

3) \( b_j = \text{likelihood } V_j \text{ in background} \)

\( a_j = " \text{ foreground} \) "

Output: Partition \( A, B \) of \( V \) s.t.

\[
\max_{A,B} g(A,B) = \sum_{i \in A} a_i + \sum_{i \in B} b_i - \sum_{(i,j) \in E} p_{ij}
\]

Change to Min Prob.

\( Q = \sum_{i} (a_i + b_i) \) then

\[
g'(A,B) = Q - \sum_{i \in A} b_i - \sum_{i \in B} a_i - \sum_{(i,j) \in E} p_{ij}
\]

Goal: \( \min g'(A,B) = \sum_{i \in A} b_i + \sum_{i \in B} a_i + \sum_{(i,j) \in E} p_{ij} \)
Claim: If $A, B$ is a cut then $s \in A \& t \in B$

$$C(A, B) = \delta'(A, B).$$

Thus, Min Cut is a Max Segmentation.
Dynamic Networks

Input: Networks $G = (V,E)$
Discrete time: $0 \leq t \leq 2$
Capacities: $C_{ij}^t = \text{cap from } a_i \text{ to } a_j \text{ at time } t.$

Example:

Buffer size: $b_t$

Idea: Make 4 copies of static network and add edges between copies.