Lecture 17:
• Network Flow
  • Choosing Good Augmenting Paths

Mar 20, 2018
Recap

• Directed graph $G = (V, E)$
• Two special nodes: source $s$ and sink $= t$
• Edge capacities $c(e)$
Recap

• An s-t flow is a function $f(e)$ such that
  - For every $e \in E$, $0 \leq f(e) \leq c(e)$ (capacity)
  - For every $v \in E$, $\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$ (conservation)
  (except $s, t$)

• The value of a flow is $\text{val}(f) = \sum_{e \text{ out of } s} f(e)$

\[
\begin{array}{c}
\text{capacity} \rightarrow 15 \\
\text{flow} \rightarrow 0
\end{array}
\]
Recap

- MaxFlow = Find an s-t flow of maximum value

\[ \text{val}(f) = 28 \]
Recap

• An \textbf{s-t cut} is a partition \((A, B)\) of \(V\) with \(s \in A\) and \(t \in B\)

• The \textbf{capacity} of a cut \((A, B)\) is \(\text{cap}(A, B) = \sum_{e \text{ out of } A} c(e)\)
Recap

• **MinCut**: Find an s-t cut of minimum capacity

\[
\text{cap}(A, B) = \sum_{e \text{ from } A \text{ to } B} c(e)
\]

• **Weak Duality Theorem**: For any flow f, cut \((A, B)\), \(\text{val}(f) \leq \text{cap}(A, B)\)

\[
\text{cap}(A, B) = 28
\]
Ford-Fulkerson Algorithm

- Start with $f(e) = 0$ for all $e \in E$
- While
  - There is an augmenting path $P$ in the residual graph $G_f$
- Augment flow along the path $P$
Ford-Fulkerson Demo

$G:$

$G_f:$
Ford-Fulkerson Demo

$G$: 

$G_f$: 

![Graph Diagram](attachment:image.png)
Ford-Fulkerson Demo

$G:$

$G_f:$
Ford-Fulkerson Demo

$G$: 

$G_f$: 

![Graph](image-url)
Ford-Fulkerson Demo

\[ A = \{s, 33\} \quad B = \{2, 5, t\} \]

\[ \text{val}(f) = 19 \quad \text{cap}(A, B) = 19 \]

So \( f \) is a max flow and \((A, B)\) is a min cut.
Summary

• The **Ford-Fulkerson Algorithm** solves maximum s-t flow
  • Running time is $O(m)$ per augmentation step
  • $O(\text{val}(f^*))$ augmentations in any graph with integer capacities
  • $O(n)$ augmentations in graphs with unit capacities

• **MaxFlow-MinCut Theorem:** The value of the max s-t flow equals the capacity of the min s-t cut
  • If $f^*$ is a max flow, the nodes reachable from $s$ in $G_{f^*}$ are a min cut
  • Given a max flow, can find a min cut in time $O(n+m)$ via BFS
Recap

• Two Problems:
  • **MaxFlow**: given a network $G=(V,E)$, capacities $c(e)$, source $s$, sink $t$, find the $s$-$t$ flow of maximum value
  • **MinCut**: given a network $G=(V,E)$, capacities $c(e)$, source $s$, sink $t$, find the $s$-$t$ cut of minimum value

• **Ford-Fulkerson Algorithm**:
  • Start with the empty flow $f(e) = 0$
  • While there is an **augmenting path** $P$ in the **residual graph** $G_f$, increase $f$ along the path
Ford-Fulkerson Algorithm

• Start with $f(e) = 0$ for all $e \in E$
• While
  • There is an augmenting path $P$ in the residual graph $G_f$
• Augment flow along the path $P$

\[
\text{val}(f^*) = 2c \\
\text{# of augmentations} = 2c
\]
Choosing Good Paths

- **Last time**: arbitrary augmenting paths
  - If FF terminates, it outputs a maximum flow

- **Today**: clever augmenting paths
  - Maximum-capacity augmenting path ("fattest path")
  - Shortest augmenting paths ("shortest path")
Fattest Augmenting Path
Fattest Augmenting Path

• Maximum-capacity augmenting path

\[
\max_{s-t \text{ paths } P} \min_{e \in P} c(e)
\]

• Can find the maximum-capacity augmenting path in time \(O(m \log n)\) using a variant of Prim’s or Kruskal’s MST
  • Exercise for the reader
### Fattest Augmenting Path

#### Arbitrary Paths
- Assume integer capacities
- Value of maxflow: \( v^* \)
- Value of aug path: \( \geq 1 \)
- Flow remaining in \( G_f \): \( \leq v^* - 1 \)
- \# of aug paths: \( \leq v^* \)

#### Maximum-Capacity Path
- Assume integer capacities
- Value of maxflow: \( v^* \)
- Value of aug path:
- Flow remaining in \( G_f \):
- \# of aug paths:
Fattest Augmenting Path

- \( f^* \) is a maximum flow with value \( v^* = val(f^*) \)
- \( P \) is a fattest augmenting s-t path with capacity \( B \)
- Claim: \( B \geq \frac{v^*}{m} \)
  
  If there is no path \( u \) with capacity \( > B \) then \( v^* \leq B \cdot m \)

- Consider \( G' \) containing only edges \( u \) with capacity \( > B \)
- s-t must be disconnected in this graph
- \( \text{cap}(A, B) \leq m \cdot B \)
- \( v^* \leq m \cdot B \)
Fattest Augmenting Path

**Arbitrary Paths**
- Assume integer capacities
- Value of maxflow: $v^*$
- Value of aug path: $\geq 1$
- Flow remaining in $G_f$: $\leq v^* - 1$
- # of aug paths: $\leq v^*$

**Maximum-Capacity Path**
- Assume integer capacities
- Value of maxflow: $v^*$
- Value of aug path: $\geq v^* / m$
- Flow remaining in $G_f$: $\leq v^* - \frac{\sqrt{e\cdot v^*}}{m}$
- # of aug paths: $\leq m \cdot \ln(v^*) = (1 - \frac{1}{m}) \cdot v^*$

After $T$ augmentations there is $v^* \cdot (1 - \frac{1}{m})^T \leq v^* \cdot e^{-T/m}$
If $T > m \cdot \ln(v^*)$ then flow remaining is $< 1$
Choosing Good Paths

• **Last time:** arbitrary augmenting paths
  • If FF terminates, it outputs a maximum flow

• **Today:** clever augmenting paths
  • Maximum-capacity augmenting path ("fattest path")
    • $\leq m \ln v^*$ augmenting paths (assuming integer capacities)
    • $O(m^2 \ln n \ln v^*)$ total running time
    • See KT for a slightly faster variant ("fat enough path")

  • Shortest augmenting paths ("shortest path")
Shortest Augmenting Path
Shortest Augmenting Path

• Find the augmenting path with the fewest hops
  • Can find shortest augmenting path in $O(m)$ time using BFS

• **Theorem:** for any capacities $\frac{nm}{2}$ augmentations suffice
  • Overall running time $O(m^2n)$  
  • Works for any capacities!

• **Warning:** proof is very tricky (you will not be tested on it)
Shortest Augmenting Path

- Let $f_i$ be the flow after the $i$-th augmenting path
- Let $G_i = G_{f_i}$ be the $i$-th residual graph
  - $f_0 = 0$ and $G_0 = G$
- Let $L_i(v)$ be the distance from $s$ to $v$ in $G_i$
  - Recall that the shortest path in $G_i$ moves layer-by-layer
Shortest Augmenting Path

• Every augmentation causes at least one edge to disappear from the residual graph, may also cause an edge to appear

• Key Property: each edge disappears at most $\frac{n}{2}$ times
  • Means that there are at most $\frac{mn}{2}$ augmentations
Shortest Augmenting Path

• **Claim 1:** for every \( v \in V \) and every \( i, L_{i+1}(v) \geq L_i(v) \)
  • Obvious for \( v = s \) because \( L_i(s) = 0 \)
  • Suppose for the sake of contradiction that \( L_{i+1}(v) < L_i(v) \)
    • Let \( v \) be the smallest such node
  • Let \( s \leadsto u \to v \) be a shortest path in \( G_{i+1} \)
    • By optimality of the path, \( L_{i+1}(v) = L_{i+1}(u) + 1 \)
    • By assumption, \( L_{i+1}(u) \geq L_i(u) \)
    • Two Cases:
      • \((u, v) \in G_i\), so \( L_i(v) \leq L_i(u) + 1 \)
        \[ L_{i+1}(v) = L_{i+1}(u) + 1 > L_i(u) + 1 > L_i(v) \]
      • \((u, v) \notin G_i\), so \((v, u)\) was in the \( i \)-th path, so \( L_i(v) = L_i(u) - 1 \)
        \[ L_{i+1}(v) = L_{i+1}(u) + 1 > L_i(u) + 1 = L_i(v) + 2 \]
Because augmenting path is the shortest path.
Shortest Augmenting Path

• Claim 2: If an edge $u \rightarrow v$ disappears from $G_i$ and reappears in $G_{j+1}$ then $L_j(u) \geq L_i(u) + 2$
  • $u \rightarrow v$ is on the $i$-th augmenting path, $L_i(v) = L_i(u) + 1$
  • $v \rightarrow u$ is on the $j$-th augmenting path, $L_j(u) = L_j(v) + 1$
  • By Claim 1: $L_j(v) \geq L_i(v)$

\[
L_j(u) = L_j(v) + 1 \geq L_i(v) + 1 = L_i(u) + 1 + 1
\]

• Every augmentation causes one edge to disappear
• At most $m$ edges, so at most $\frac{mn}{2}$ disappearances
• At most $\frac{mn}{2}$ augmenting paths
Choosing Good Paths

• **Last time:** arbitrary augmenting paths
  • If FF terminates, it outputs a maximum flow

• **Today:** clever augmenting paths
  • Maximum-capacity augmenting path ("fattest path")
    • \( \leq m \ln v^* \) augmenting paths (assuming integer capacities)
    • \( O(m^2 \ln n \ln v^*) \) total running time
    • See KT for a slightly faster variant ("fat enough path")

• Shortest augmenting paths ("shortest path")
  • \( \leq \frac{mn}{2} \) augmenting paths (for any capacities)
  • \( O(m^2 n) \) total running time
Summary

• The Ford-Fulkerson Algorithm solves maximum s-t flow
  • Different choices of augmenting paths give different running times

• Still an active area of research!

\[ O(m^{2/5}n^{4/5}\log^2 \sqrt{n}) \quad O(m^{10/7}) \]

• Can solve many problems efficiently via reductions to the maximum flow or minimum cut problems