The Network Flow Problem

How much stuff can flow from \( s \) to \( t \)?

Net Flow: Formal Definition

Given:
- A digraph \( G = (V,E) \)
- Two vertices \( s, t \) in \( V \) (source & sink)
- A capacity \( c(u,v) \geq 0 \) for each \( (u,v) \in E \) (and \( c(u,v) = 0 \) for all non-edges \( (u,v) \))

Find:
- A flow function \( f: V \times V \to \mathbb{R} \) s.t.,
  - \( f(u,v) \leq c(u,v) \) [Capacity Constraint]
  - \( f(u,v) = -f(v,u) \) [Skew Symmetry]
  - \( f(u,V) = 0 \) for all \( u \neq s,t \) [Flow Conservation]

Maximizing total flow \( |f| = f(s,V) \)

Notation:
- \( f(x,y) = \sum_{x \in X} \sum_{y \in Y} f(x,y) \)

Example: A Flow Function

\[
\begin{align*}
    f(s,u) &= f(u,t) = 2 \\
    f(u,s) &= f(t,u) = -2 \quad \text{(Why?)} \\
    f(s,t) &= -f(t,s) = 0 \quad \text{(In every flow function for this \( G \). Why?)} \\
    f(u,V) &= \sum_{v \in V} f(u,v) = f(u,s) + f(u,t) = -2 + 2 = 0
\end{align*}
\]

Max Flow via a Greedy Alg?

While there is an \( s \to t \) path in \( G \)
- Pick such a path, \( p \)
- Find \( c_p \), the min capacity of any edge in \( p \)
- Subtract \( c_p \) from all capacities on \( p \)
- Delete edges of capacity 0

- Not shown: \( f(u,v) \) if \( \leq 0 \)
- Note: max flow \( \geq 4 \) since \( f \) is a flow function, with \( |f| = 4 \)
Max Flow via a Greedy Alg?

This does NOT always find a max flow:
If you pick $s \rightarrow b \rightarrow a \rightarrow t$ first,

\[
\begin{array}{c}
\text{Flow stuck at 2. But flow 3 possible.}
\end{array}
\]

A Brief History of Flow

<table>
<thead>
<tr>
<th>Year</th>
<th>Author(s)</th>
<th>Conference/Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Goldberg &amp; Rao</td>
<td>FOCS '97</td>
</tr>
<tr>
<td>2007</td>
<td>King &amp; Lue</td>
<td>Proc. of SFCS 2007</td>
</tr>
<tr>
<td>1997</td>
<td>King &amp; Lue</td>
<td>Proc. of SFCS 1997</td>
</tr>
<tr>
<td>1996</td>
<td>King &amp; Lue</td>
<td>Proc. of SFCS 1996</td>
</tr>
<tr>
<td>1994</td>
<td>King &amp; Lue</td>
<td>Proc. of SFCS 1994</td>
</tr>
<tr>
<td>1991</td>
<td>King &amp; Lue</td>
<td>Proc. of SFCS 1991</td>
</tr>
<tr>
<td>1988</td>
<td>King &amp; Lue</td>
<td>Proc. of SFCS 1988</td>
</tr>
</tbody>
</table>

$n = \# \text{ of vertices}$
$m = \# \text{ of edges}$
$U = \text{Max capacity}$

Residual Capacity

- The residual capacity (w.r.t. $f$) of $(u,v)$ is $c_f(u,v) = c(u,v) - f(u,v)$
- E.g.:
  - $c_f(s,b) = 7$;
  - $c_f(a,x) = 1$;
  - $c_f(x,a) = 3$;
  - $c_f(x,t) = 0$ (a saturated edge)

Residual Networks & Augmenting Paths

- The residual network (w.r.t. $f$) is the graph $G_f = (V,E_f)$, where $E_f = \{ (u,v) \mid c_f(u,v) > 0 \}$
- An augmenting path (w.r.t. $f$) is a simple $s \rightarrow t$ path in $G_f$.

A Residual Network

residual network: the graph $G_f = (V,E_f)$, where $E_f = \{ (u,v) \mid c_f(u,v) > 0 \}$
An Augmenting Path

Lemma 1

If \( f \) admits an augmenting path \( p \), then \( f \) is not maximal.

Proof: "obvious" -- augment along \( p \) by \( c_p \), the min residual capacity of \( p \)'s edges.

Augmenting A Flow

Proof of Lemma 1—Case 1

Let \((u,v)\) be any edge in augmenting path. Note

\[ c(u,v) = c(u,v) - \hat{f}(u,v) \geq c_p > 0 \]

Case 1: \( \hat{f}(u,v) \geq 0 \):

Add forward flow

Proof of Lemma 1—Case 2

Let \((u,v)\) be any edge in augmenting path. Note

\[ c(u,v) = c(u,v) - \hat{f}(u,v) \geq c_p > 0 \]

Case 2: \( \hat{f}(u,v) \leq -c_p \):

Cancel/redirec reverse flow

Proof of Lemma 1—Case 3

Let \((u,v)\) be any edge in augmenting path. Note

\[ c(u,v) = c(u,v) - \hat{f}(u,v) \geq c_p > 0 \]

Case 3: \(-c_p \leq \hat{f}(u,v) < 0\):

???
Proof of Lemma 1—Case 3

Let \((u,v)\) be any edge in augmenting path. Note
\[
c_f(u,v) = c(u,v) - f(u,v) > 0
\]

Case 3: \(-c_p \leq f(u,v) < 0\)
\[
c_p \geq f(v,u) > 0
\]

Both:
cancel/redirect reverse flow and add forward flow

Ford-Fulkerson Method

While \(G_f\) has an augmenting path, augment

Questions:
- Does it halt?
- Does it find a maximum flow?
- How fast?

Max Flow / Min Cut Theorem

For any flow \(f\), the following are equivalent
1. \(|f| = c(S,T)\) for some cut \(S,T\) (a min cut)
2. \(f\) is a maximum flow
3. \(f\) admits no augmenting path

Proof:
1. \(\Rightarrow\) 2: corollary to lemma 2
2. \(\Rightarrow\) 3: contrapositive of lemma 1

Cuts

- A partition \(S,T\) of \(V\) is a cut if \(s \in S, t \in T\)
- Capacity of cut \(S,T\) is \(c(S,T) = \sum_{u \in S} c(u,v) + \sum_{v \in T} c(u,v)\)

Lemma 2

- For any flow \(f\) and any cut \(S,T\),
  - the net flow across the cut equals the total flow, i.e., \(|f| = f(S,T)\), and
  - the net flow across the cut cannot exceed the capacity of the cut, i.e. \(f(S,T) \leq c(S,T)\)
- Corollary:
  Max flow \(\leq\) Min cut

Max Flow / Min Cut Theorem

For any flow \(f\), the following are equivalent
1. \(|f| = c(S,T)\) for some cut \(S,T\) (a min cut)
2. \(f\) is a maximum flow
3. \(f\) admits no augmenting path

Proof:
1. \(\Rightarrow\) 2: corollary to lemma 2
2. \(\Rightarrow\) 3: contrapositive of lemma 1

(3) \(\Rightarrow\) (1)

\(S = \{ u \mid \exists\text{ an augmenting path from } s \text{ to } u \}\)
\(T = V - S; s \in S, t \in T\)
For any \((u,v)\) in \(S \times T\), \(\exists\) an augmenting path from \(s\) to \(u\), but not to \(v\).
\(- (u,v)\) has 0 residual capacity:
\((u,v) \in E \Rightarrow \text{saturated} \quad f(u,v) = c(u,v)\)
\((v,u) \in E \Rightarrow \text{no flow} \quad f(u,v) = 0 = -f(v,u)\)
This is true for every edge crossing the cut, i.e.
\(|f| = f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) = \sum_{u \in S} c(u,v) = c(S,T)\)
Corollaries & Facts

- If Ford-Fulkerson terminates, then it's found a max flow.
- It will terminate if $c(e)$ integer or rational (but may not if they're irrational).
- However, may take exponential time, even with integer capacities:

$\begin{array}{ccc}
\text{s} & \text{c} & \text{t} \\
\text{a} & \text{c} & \text{b} \\
\text{c} & \text{c} & \text{c} \\
\text{c} = 10^{99}, \text{say}
\end{array}$

Edmonds-Karp Algorithm

- Use a shortest augmenting path (via Breadth First Search in residual graph)
- Time: $O(n m^2)$

BFS/Shortest Path Lemmas

Distance from $s$ is never reduced by:
- Deleting an edge
  - proof: no new (hence no shorter) path created
- Adding an edge $(u,v)$, provided $v$ is nearer than $u$
  - proof: BFS is unchanged, since $v$ visited before $(u,v)$ examined

Lemma 3

Let $f$ be a flow, $G_f$ the residual graph, and $p$ a shortest augmenting path. Then no vertex is closer to $s$ after augmentation along $p$.

Proof: Augmentation only deletes edges, adds back edges

Augmentation vs BFS

The Edmonds-Karp Algorithm performs $O(mn)$ flow augmentations

Proof:
- $(u,v)$ is critical on augmenting path $p$ if it's closest to $s$ having min residual capacity.
- Won't be critical again until farther from $s$.
- So each edge critical at most $n$ times.
Augmentation vs BFS Level

Corollary

Edmonds-Karp runs in $O(nm^2)$

Flow Integrality Theorem

If all capacities are integers
- The max flow has an integer value
- Ford-Fulkerson method finds a max flow in which $f(u,v)$ is an integer for all edges $(u,v)$

Bipartite Maximum Matching

Bipartite Graphs:
- $G = (V,E)$
- $V = L \cup R$ ($L \cap R = \emptyset$)
- $E \subseteq L \times R$

Matching:
- A set of edges $M \subseteq E$ such that no two edges touch a common vertex

Problem:
- Find a matching $M$ of maximum size

Reducing Matching to Flow

Given bipartite $G$, build flow network $N$ as follows:
- Add source $s$, sink $t$
- Add edges $s \to L$
- Add edges $R \to t$
- All edge capacities 1

Theorem:
Max flow iff max matching

Reducing Matching to Flow

Theorem: Max matching size = max flow value

$M \to f$? Easy – send flow only through $M$
$f \to M$? Flow integrality Thm, + cap constraints
Notes on Matching

- Max Flow Algorithm is probably overly general here
- But most direct matching algorithms use "augmenting path" type ideas similar to that in max flow – See text & homework
- Time $mn^{1/2}$ possible via Edmonds-Karp