Review: Augmenting Flows

residual graph; edges: forward (difference), reverse (existing flow)
augmenting path: \( s \rightarrow t \) in residual graph, bottleneck capacity

Step 3: F-F returns a maximum flow

We will prove this by establishing a deep connection between flows and cuts in graphs: the max-flow min-cut theorem.

- An \( s-t \) cut \( (A, B) \) is a partition of the nodes into sets \( A \) and \( B \) where \( s \in A \), \( t \in B \)
- Capacity of cut \( (A, B) \) equals
  \[
  c(A, B) = \sum_{e \text{ from } A \text{ to } B} c(e)
  \]
- Flow across a cut \( (A, B) \) equals
  \[
  f(A, B) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)
  \]

Example of Cut

Exercise: write capacity of cut and flow across cut.
Capacity is 29 and flow across cut is 19.

Clicker Question
What is the capacity of the cut and the flow across the cut?

<table>
<thead>
<tr>
<th>Capacity</th>
<th>Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. 16+4+9+14</td>
<td>11+1+3+11</td>
</tr>
<tr>
<td>B. 16+4-9+14</td>
<td>11+1-4+11</td>
</tr>
<tr>
<td>C. 16+4+14</td>
<td>11+1-4+11</td>
</tr>
<tr>
<td>D. 16+4+14</td>
<td>11+1+1+1</td>
</tr>
</tbody>
</table>
Flow Value Lemma

First relationship between cuts and flows

**Lemma:** let $f$ be any flow and $(A, B)$ be any $s$-$t$ cut. Then

$$v(f) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$$

**Proof** (see book) use conservation of flow:
all the flow out of $s$ must leave $A$ eventually.
Rewrite flow as $v(f) = \sum_{v \in A} f^{	ext{out}}(v) - f^{	ext{in}}(v)$
only nonzero difference is $f(s)$
Consider cases: edge in $A$, leading out of $A$, leading into $A$

Duality: Max Flow – Min Cut

**Claim** If there is a flow $f^*$ and cut $(A^*, B^*)$ such that $v(f^*) = c(A^*, B^*)$, then

- $f^*$ is a maximum flow
- $(A^*, B^*)$ is a minimum cut

Corollary: Cuts and Flows

Really important corollary of flow-value lemma

**Corollary:** Let $f$ be any $s$-$t$ flow and let $(A, B)$ be any $s$-$t$ cut. Then $v(f) \leq c(A, B)$.

**Proof:**

$$v(f) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$$

$$\leq \sum_{e \text{ out of } A} c(e)$$

$$= c(A, B)$$

Clicker

Suppose $f$ is a flow, and there is a path from $s$ to $u$ in $G_f$, but no path from $s$ to $v$ in $G_f$. Then

- A. There is no edge from $u$ to $v$ in $G$.
- B. If there is an edge from $u$ to $v$ in $G$ then $f$ does not send any flow on this edge.
- C. If there is an edge from $u$ to $v$ in $G$ then $f$ fully saturates it with flow.
- D. None of the above.

F-F returns a maximum flow

**Theorem:** The $s$-$t$ flow $f$ returned by F-F is a maximum flow.

- Since $f$ is the final flow there are no residual paths in $G_f$.
- Let $(A, B)$ be the $s$-$t$ cut where $A$ consists of all nodes reachable from $s$ in the residual graph.
  - Any edge out of $A$ must have $f(e) = c(e)$ otherwise there would be more nodes than just $A$ that reachable from $s$.
  - Any edge into $A$ must have $f(e) = 0$ otherwise there would be more nodes than just $A$ that reachable from $s$.
- Therefore

$$v(f) = \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ into } A} f(e)$$

$$= \sum_{e \text{ out of } A} c(e) = c(A, B)$$
**F-F finds a minimum cut**

**Theorem:** The cut \((A, B)\) where \(A\) is the set of all nodes reachable from \(s\) in the residual graph is a minimum-cut.

![Graph](image)

**Ford-Fulkerson Running Time**

- Flow increases at least one unit per iteration
- F-F terminates in at most \(C_s\) iterations, where \(C_s\) is sum of capacities leaving source.
- \(C_s \leq n C_{\text{max}}\) (in terms of maximum edge capacity)
- Running time: \(O(m n C_{\text{max}})\)

Is this polynomial? *pseudo-polynomial* (exponential in \(\log C_{\text{max}}\))

**Improving Running Time**

Good path choice will find:
- \(s \to u \to t\), flow \(C\)
- \(s \to v \to t\), flow \(C\)

Worst-case: keep incrementing by 1:
- \(s \to u \to v \to t\), flow 1
- \(s \to v \to u \to t\), flow 1
- \(s \to u \to v \to t\), flow 1
- ...

Solution: choose good augmenting paths, with
- Large enough bottleneck capacity: *capacity-scaling algorithm*
- Fewest edges: Edmonds-Karp, Dinitz

**Capacity-scaling algorithm**

Idea: ignore edges with small capacity at first

Original residual graph \(G_f\)

\[ G_f(\Delta) \text{ for } \Delta = 100. \text{ Def: only edges with residual capacity } \geq \Delta \]
Capacity-scaling algorithm

Start with large $\Delta$, divide by two in each phase

let $f(e) = 0$ for all $e \in E$

let $\Delta = \text{largest power of 2 \leq } C_{\max}$

while $\Delta \geq 1$

prune residual graph $G_f$ to $G_f(\Delta)$

while there is augmenting $s \rightarrow t$ path $P$ in $G_f(\Delta)$ do

$f = \text{Augment}(f, P)$

update $G_f(\Delta)$ $\triangleright$ only $c_e \geq \Delta$

end while

$\Delta = \Delta/2$ $\triangleright$ refine

end while

Capacity-Scaling: Running Time

▶ How many scaling phases? $\Theta(\log C_{\max})$

▶ How much does the flow increase at every augmentation? $\geq \Delta$

▶ How many augmentations per phase? $\leq 2m$

▶ Can show: at end of $\Delta$ phase, flow value within $m\Delta$ of max.

$\implies$ at most $2m$ iterations $\Delta/2$ phase

▶ (Sketch) Construct cut $(A, B)$ as in max-flow / min-cut theorem.

▶ Edges from $A$ to $B$ are within $\Delta$ of being saturated.

▶ Edges from $B$ to $A$ carry less than $\Delta$ flow.

$\implies$ Cut capacity at most $m\Delta$ more than flow value.

▶ Recall: time to find augmenting path? $O(m)$

▶ Overall: $O(m^2 \log C_{\max})$, polynomial

Choosing Short Augmenting Paths

Two similar algorithms: Edmonds-Karp, Dinitz

▶ Work as usual on residual graph

▶ Use BFS from $s$ to construct level graph

keep only edges from level $k$ to $k+1$

$\Rightarrow$ force choosing shortest augmenting paths

▶ Each new augmenting path removes one bottleneck edge

at most $m$ augmentations per phase (no back edges)

when level graph disconnected, must consider longer paths

▶ Construct new level graph (new BFS)

at most $n-1$ different lengths $\Rightarrow < n$ phases

▶ Complexity: $O(m^2n)$; polynomial, capacity-independent

More intricate variant (Dinitz) achieves $O(mn^2)$

Running Times

▶ Basic F-F: $O(mnC_{\max})$ pseudo-polynomial

▶ polynomial in magnitude

▶ Capacity-scaling: $O(m^2 \log C_{\max})$ polynomial

▶ polynomial in number of bits

▶ Edmonds-Karp: $O(m^2n)$ strongly-polynomial

▶ does not depend on values, only $m, n$

▶ Dinitz: $O(mn^2)$ even better