A Puzzle

How many loads of grain can you ship from $s$ to $t$? Which boats are used?

Input: Flow Network

Problem input is a flow network:
- Directed graph
- Source node $s$
- Target node or sink $t$
- Edge capacities $c(e) \geq 0$

Solution: A Flow

A network flow is an assignment of values $f(e)$ to each edge $e$, which satisfy:
- Capacity constraints: $0 \leq f(e) \leq c(e)$ for all $e$
- Flow conservation:
  $$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$
  for all $v \notin \{s, t\}$.
- Value $v(f)$ of flow $f =$ total flow on edges leaving source
- Max flow problem: find a flow of maximum value

Algorithm Design Techniques

- Greedy
- Divide and Conquer
- Dynamic Programming
- Network Flows
Network Flow

- Previous topics (Greedy, Divide-and-Conquer, Dynamic Programming) were design techniques.
- Network flow relates to a specific class of problems with many applications.

- Direct applications: commodities in networks
- Indirect applications: Matching in graphs
- Airline scheduling
- Baseball elimination

Plan: design and analyze algorithms for max-flow problem, then apply to solve other problems.

Designing a Max-Flow Algorithm

First idea: initialize to zero flow and then repeatedly “augment” flow on paths from $s$ to $t$ until we can no longer do so.

Problem: we are now stuck. All paths from $s$ to $t$ have a saturated edge.

We would like to “augment” $s \xrightarrow{1} v \xleftarrow{1} u \xrightarrow{1} t$, but this is not a real $s \rightarrow t$ path. How can we identify such an opportunity?

Exercise: draw the residual graph

Residual Graph

The residual graph $G_f$ identifies opportunities to increase flow on edges with leftover capacity, or decrease flow on edges already carrying flow.

For each original edge $e = (u, v)$ in $G$, it has:

- A forward edge $e' = (u, v)$ with residual capacity $c(e) - f(e)$
- A reverse edge $e' = (v, e)$ with residual capacity $f(e)$

Edges with zero residual capacity are omitted.

Augment Operation

Revised Idea: use paths in the residual graph to augment flow.

$P = s \rightarrow v \rightarrow u \rightarrow t$ has bottleneck capacity 1.

- Increase flow for forward edges, decrease for backward edges.
- Augment $s \xrightarrow{1} v \xleftarrow{1} u \xrightarrow{1} t$
Augment Operation

Revised Idea: use paths in the residual graph to augment flow

Augment($f, P$)
Let $b = \text{bottleneck}(P, f)$ \hspace{1cm} $\triangleright$ least residual capacity in $P$
for each edge $(u, v)$ in $P$ do
  if $(u, v)$ is a forward edge then
    Let $e = (u, v)$ be the original edge
    $f(e) = f(e) + b$ \hspace{1cm} $\triangleright$ increase flow on forward edges
  else $(u, v)$ is a backward edge
    Let $e = (v, u)$ be the original edge
    $f(e) = f(e) - b$ \hspace{1cm} $\triangleright$ decrease flow on backward edges
  end if
end for

Ford-Fulkerson Algorithm

Repeatedly find augmenting paths in the residual graph and use them to augment flow!

Ford-Fulkerson($G, s, t$)
\hspace{1cm} $\triangleright$ Initially, no flow
Initialize $f(e) = 0$ for all edges $e$
Initialize $G_f = G$
\hspace{1cm} $\triangleright$ Augment flow as long as it is possible
while there exists a $s$-$t$ path $P$ in $G_f$ do
  $f = \text{Augment}(f, P)$
  update $G_f$
end while
return $f$

Ford-Fulkerson Analysis

$\triangleright$ Step 1: argue that F-F returns a flow
$\triangleright$ Step 2: analyze termination and running time
$\triangleright$ Step 3: argue that F-F returns a maximum flow
Step 1: F-F returns a flow

Claim: If $f$ is a flow then $f' = \text{Augment}(f, P)$ is also a flow.

Proof idea. Verify two conditions for $f'$ to be a flow: capacity and flow conservation.

Capacity

- Suppose original edge is $e = (u, v)$
- If $e$ appears in $P$ as a forward edge $(u \xrightarrow{b} v)$, then flow increases by bottleneck capacity $b$, which is at most $c(e) - f(e)$, so does not exceed $c(e)$.
- If $e$ appears in $P$ as a reverse edge $(v \xleftarrow{b} u)$, then flow decreases by bottleneck capacity $b$, which is at most $f(e)$, so is at least 0.

Flow Conservation

- Consider any node $v$ in the augmenting path, and do a case analysis on the types of the incoming and outgoing edge:

  residual graph: $P = s \leadsto u \rightarrow v \rightarrow w \leadsto t$

  original graph: $u \xrightarrow{b} v \xrightarrow{b} w$

  $u \xrightarrow{b} v \xleftarrow{b} w$

  $w \xleftarrow{b} u \xrightarrow{b} w$

  $u \xleftarrow{b} v \xleftarrow{b} w$

- In all cases, the change in incoming flow to $v$ is equal to the change in outgoing flow from $v$.

Step 2: Termination and Running Time

Assumption: All capacities are integers. By nature of F-F, all flow values and residual capacities remain integers during the algorithm.

Running time:

- In each F-F iteration, flow increases by at least 1. Therefore, number of iterations is at most $v(f^*)$, where $f^*$ is the final flow.
- Let $C$ be the total capacity of edges leaving source $s$
- Then $v(f^*) \leq C$.
- So F-F terminates in at most $C$ iterations

Running time per iteration? $O(m + n)$ to find an augmenting path