Lecture 19:
• The Max-Flow Problem

Announcements
• HW-4 due, Project Step 2 due
• PS-5 out, due Tues 11/8
• HW-5 out (soon)

Max-Flow Intro

The “Lucky Puck” Company Flow Network (Cormen et al)

• Vancouver factory (source) truck routes to Winnipeg warehouse (target)
• Company ships puck crates through intermediate cities each day
• Crates can not be stored at the intermediate cities
• Edge labels are max crates per day that can be shipped between cities

The Max-Flow Problem: What is the maximum flow for the network?
Max-Flow Intro

Other Examples
• liquids flowing through pipes
• parts flowing through assembly lines
• current flowing through electrical networks
• information flowing through communication networks

“Flow Conservation” property
• rate material enters a vertex must equal rate it leaves the vertex

\[ v_1 \quad f_1 \quad v_2 \quad f_2 \quad v_3 \]

• \( f_1 \) and \( f_2 \) are capacity constraints ... \( v_2 \) cannot “store” material
• if \( f_1 > f_2 \), then \( v_1 \) can only send \( f_2 \) rate to \( v_2 \)
• if \( f_1 < f_2 \), then \( v_2 \) can only send \( f_1 \) rate to \( v_3 \)

A flow network \( G = (V, E) \) is a directed graph such that:
• each edge \((u, v) \in E\) has a capacity \( c(u, v) \geq 0 \) ... nonnegative
• \( G \) has distinguished vertices \( s \) and \( t \) ... source and target
• \( f : V \times V \rightarrow \mathbb{R} \) represents a “flow” ... with constraints

A flow function \( f \) for \( G \) must satisfy:

Capacity constraint: For \( u, v \in V \), requires \( 0 \leq f(u, v) \leq c(u, v) \)

Flow conservation: For \( u \in V - \{s, t\} \), requires \( \sum_{v \in V} f(v, u) = \sum_{w \in V} f(u, w) \)

Additional constraints on flow networks:
• assume each vertex lies on a path from \( s \) to \( t \)
• if \((u, v) \in E\) then no edge \((v, u)\) in opposite direction ... can add node
• if \((u, v) \not\in E\) then \( c(u, v) = 0 \)
• \( G \) does not have any self-edges
Max-Flow Intro

The value $|f|$ of a flow $f$ is:

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

- note: networks typically don’t have edges into $s$ ... so $\sum f(v, s) = 0$
- if no edges into $s$, the value of a flow is what flows out of $s$

Thus, the Max-Flow problem (more formally) is:

Given $G$, $s$, $t$, and $c$, find the flow $f$ with the largest possible value $|f|$

Ford-Fulkerson Method

The Basic Idea:

1. initialize flow $f$ to 0
2. while an augmenting path $p$ exists in the residual network $G_f$
3. augment flow $f$ along the path $p$
4. return $f$

To develop a concrete algorithm, need to define:

- residual networks
- augmenting paths
Residual Networks

An edge’s “**residual capacity**” $c_f = c(u, v) - f(u, v)$ (capacity minus flow)
- if positive, edge’s flow can be increased
- if zero, edge is at max capacity
- if negative, edge is over capacity (violates network constraints)

A **residual network** $G_f$ captures how edge flow can be changed in $G$

(i) $G_f$ contains edges of $G$ with a positive residual capacity $c_f$
(ii) $G_f$ can contain edges not in $G$ that represent a decrease in positive flow
    - i.e., for $f(u, v)$, place edge $(v, u)$ in $G_f$ with $c_f(v, u) = f(u, v)$
    - where decreasing flow on edges could help to increase the total flow
    - we are “sending back” flow that has already been sent along an edge