1 Basics

Flow. The *capacity* indicates how much flow can be allowed on an edge. Given a directed graph with edge capacity $c(u, v)$ and s, t, a flow is a mapping $f : E \to \mathbb{R}^+$ that satisfies

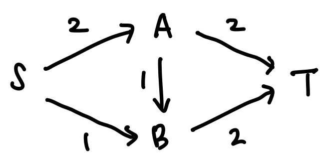
- Capacity constraint: $f(u, v) \le c(u, v)$, the flow on an edge cannot exceed its capacity.
- Conservation of flows: $f^{\text{in}}(v) = f^{\text{out}}(v)$, flow in equals flow out for any $v \notin \{s, t\}$

Here, we define $f^{\text{in}}(v) = \sum_{u:(u,v)\in E} f(u, v)$ and $f^{\text{out}}(v) = \sum_{u:(v,u)\in E} f(u, v)$. We also define $f(v, u) = -f(u, v)$, and this is called *skew-symmetry*. Note that the total flow in the graph is $\sum_{v:(s,v)\in E} f(s, v) = \sum_{u:(u,t)\in E} f(u, t)$, where s is the source node of the graph and t is the target node.

Residual Graph. Given a flow network (G, s, t, c) and a flow f, the *residual capacity* (w.r.t. flow f) is denoted by $c_f(u, v) = c_{uv} - f_{uv}$. And the *residual network* $G_f = (V, E_f)$ where $E_f = \{(u, v) : c_f(u, v) > 0\}$.

Ford-Fulkerson. Keep pushing along s-t paths in the residual graph and update the residual graph accordingly. Runs in time $O(mF)$.

starting network:

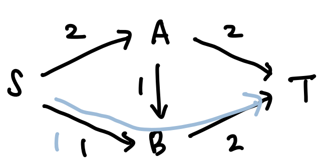


each edge weight denotes the capacity of each edge

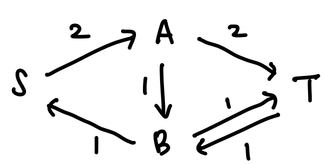
Question: what's the max flow we can push S → T ?

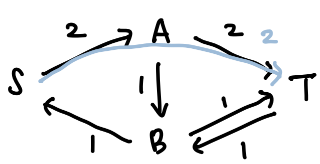
residual graph: edge weights denote how much more flow we can push along that edge (remaining capacity)

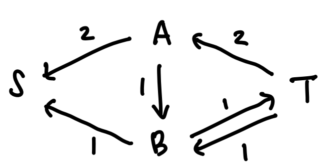
flow:

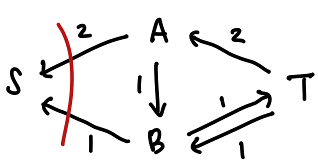


residual graph:









min cut: 3

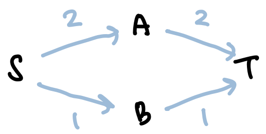
no more s-t paths

max flow: 3

<u>min cut</u> : "bottle neck" of the network

* min cut = max flow

final flow $s \to t$ from algorithm above:



conservation of flow: flow into node = flow out of node

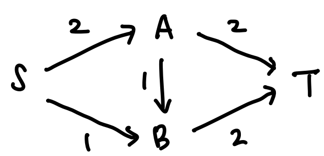
$$f^{in}(v) = f^{out}(v) \quad \forall v \notin \{s,t\}$$

capacity constraint : flow along edge cannot exceed capacity

$$f(u,v) \leq c(u,v) \quad \text{for all edges } (u,v)$$

solving max flow with LP:

*by conservation of flow, flow out of s equals flow into t



max $f_{SA} + f_{SB}$ (alt: max $f_{AT} + f_{BT}$ equivalent by conservation of flow)

s.t. $f_{SA} - f_{AB} - f_{AT} = 0$

$f_{SB} - f_{BT} = 0$

$f_{SA} \leq 2$

$f_{AB} \leq 1$

$f_{AT} \leq 2$

$f_{SB} \leq 1$

$f_{BT} \leq 2$

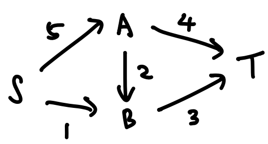
$f_{SA}, f_{AB}, f_{AT}, f_{SB}, f_{BT} \geq 0$

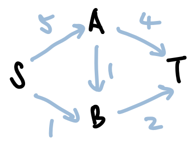
2 Max-Flow Min Cut Basics

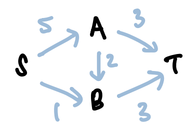
For each of the following, state whether the statement is True or False. If true provide a short proof, if false give a counterexample.

(a) If all edge capacities are distinct, the max flow is unique.

False





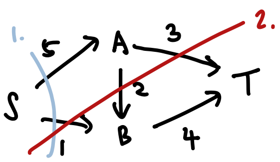


Flow can split along paths, doesn't have to saturate capacities.

(b) If all edge capacities are distinct, the min cut is unique.

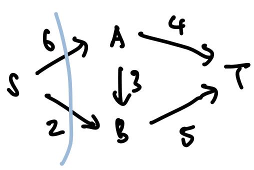
False

Could have multiple bottlenecks.



(c) If all edge capacities are increased by an additive constant, the min cut remains unchanged.

False, cuts containing more edges will increase more in value.



(d) If all edge capacities are multiplied by a positive integer, the min cut remains unchanged.

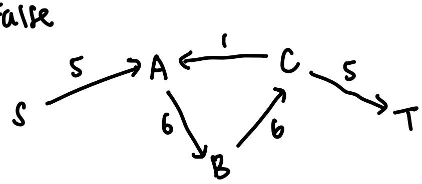
True

$C_{e'}$ = min cut $\quad$ C_e = some other cut

$\sum_e C_{e'} \le \sum_e C_e$ $\quad$ since $a > 0$, $a \sum_e C_{e'} \le a \sum_e C_e$

(e) In any max flow, there is no directed cycle on which every edge carries positive flow.

False



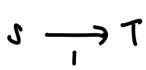
arrows show the actual flow

conservation of flow ✓

whirlpool in the middle

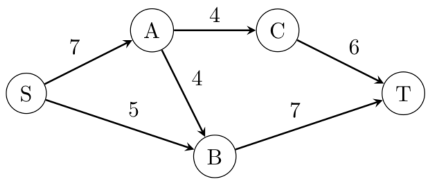
(f) There exists a max flow for which there is no directed cycle on which every edge carries positive flow.

True

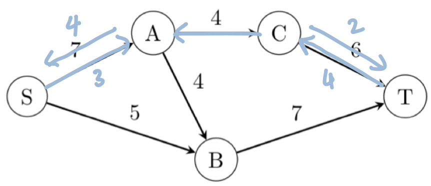


3 Residual in graphs

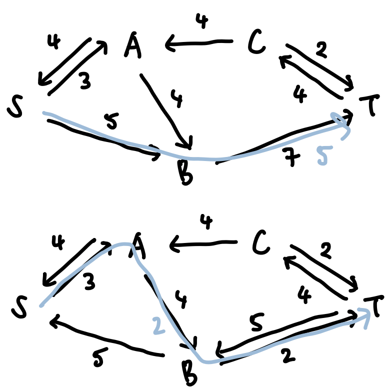
Consider the following graph with edge capacities as shown:

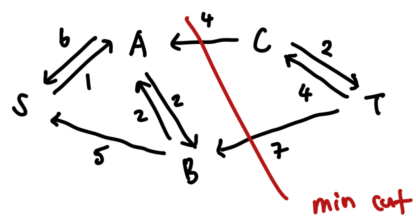


(a) Consider pushing 4 units of flow through $S \to A \to C \to T$. Draw the residual graph after this push.



(b) Compute a maximum flow of the above graph. Find a minimum cut. Draw the residual graph of the maximum flow.





no more flow possible to push

max flow : 11

4 Secret Santa (Challenge Problem)

Imagine you are throwing a party and you want to play Secret Santa. Thus you would like to assign to every person at the party another partier to whom they must anonymously give a single gift. However, there are some restrictions on who can give gifts to who: nobody should be assigned to give a gift to themselves or to their spouse. Since you are the host, you know all of these restrictions. Give an efficient algorithm that determines if you and your guests can play Secret Santa.

Duality

primal problem from last week:

$$\min \quad 10x_1 + 20x_2 = OPT_p \qquad\qquad \min \quad \begin{bmatrix} 10 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad c^T x$$

$$\text{s.t.} \quad x_1 + 2x_2 \geq 3 \qquad \xrightarrow[\text{form}]{\text{matrix}} \qquad \begin{bmatrix} 1 & 2 \\ 4 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 3 \\ 2 \end{bmatrix} \qquad Ax \geq b$$

$$4x_1 - x_2 \geq 2$$

$$x_1, x_2 \geq 0 \qquad\qquad x_1, x_2 \geq 0 \qquad x \geq 0$$

Dual:

- motivation: suppose it's difficult to find the absolute minimum for primal object, we can perhaps prove a lower bound "weak duality"

  show that $OPT_p \geq$ constant $\leftarrow OPT_d$

- can also be used to prove optimality of the primal

  if $OPT_p = K$, and we show $OPT_d = K$, it means primal can't possibly be smaller than K, so K is the minimum "strong duality"

$$\min \quad 10x_1 + 20x_2$$

$$y_1(x_1 + 2x_2) \geq 3y_1$$

$$y_2(4x_1 - x_2) \geq 2y_2$$

$$\underset{\leq 10}{(y_1 + 4y_2)}x_1 + \underset{\leq 20}{(2y_1 - y_2)}x_2 \geq \underbrace{3y_1 + 2y_2}_{\text{lower bound to maximize}}$$

$$\max \quad 3y_1 + 2y_2 = OPT_d \qquad\qquad \max \quad \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \qquad b^T y$$

$$\text{s.t.} \quad y_1 + 4y_2 \leq 10 \qquad\qquad \text{s.t.} \quad \begin{bmatrix} 1 & 4 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \leq \begin{bmatrix} 10 \\ 20 \end{bmatrix} \qquad A^T y \leq c$$

$$2y_1 - y_2 \leq 20 \qquad \xrightarrow[\text{form}]{\text{matrix}}$$

$$y_1, y_2 \geq 0 \qquad\qquad y \geq 0$$

Thm: weak duality

$OPT_p \geq OPT_d$ when primal is minimization

$OPT_p \leq OPT_d$ when primal is maximization

Thm: strong duality

If primal is bounded and feasible $\Rightarrow OPT_p = OPT_d$

tinyurl.com/disc8cs170

5 Taking a Dual

Consider the following linear program:

$$\begin{aligned} \max\ 4x_1 + 7x_2 \\ x_1 + 2x_2 \leq 10 \\ 3x_1 + x_2 \leq 14 \\ 2x_1 + 3x_2 \leq 11 \\ x_1, x_2 \geq 0 \end{aligned}$$

Construct the dual of the above linear program.

$y_1(x_1 + 2x_2) \leq 10y_1$

$y_2(3x_1 + x_2) \leq 14y_2$

$y_3(2x_1 + 3x_2) \leq 11y_3$

combine like terms of x_1, x_2

$$\underbrace{(y_1 + 3y_2 + 2y_3)}_{\geq 4} x_1 + \underbrace{(2y_1 + y_2 + 3y_3)}_{\geq 7} x_2 \leq \underbrace{10y_1 + 14y_2 + 11y_3}_{\text{min this upper bound}}$$

$$\left.\begin{aligned} \min\ & 10y_1 + 14y_2 + 11y_3 \\ \text{s.t.}\ & y_1 + 3y_2 + 2y_3 \geq 4 \\ & 2y_1 + y_2 + 3y_3 \geq 7 \\ & y_1, y_2, y_3 \geq 0 \end{aligned}\right\} \text{Dual}$$