Two Fundamental Concepts

15-451
9/10/20

Cuts and Flows

A flow is a "material" "flowing" through a "medium".

Examples from Physics

Water in: 1) Stream
2) Atmosphere
3) Pipe
4) Earth

Thermal Heat in:
1) Computer chip.
2) Jet engine
3) Cosmos

Electricity in: 1) wire
2) Cosmos

Examples from Operations Research (OR)

1) Moving goods to market
2) Matching jobs to resourses.

Question: What laws govern these flows!

In this class we consider flows on edges of a Graph.

Such that: 1) the flow into a vertex equals
" "out of" ".

2) The flow on an edge does not exceed the edge capacity.

The Maximum Flow Prob

Def A <u>flow network</u> is

1) $G=(V,E)$ directed (oriented)
2) Edge capacities $c: V \times V \to \mathbb{R}$
   st $c(u,v) \geq 0$
   a) $(u,v) \in E$ then $c(u,v), c(v,u) \geq 0$ (formally)
   or b) $c(u,v) > 0$ & $c(v,u) = 0$ (ok)

   eg $(u,v) \notin E$ then $c(u,v) = 0$
3) $s \neq t \in V$ $s \equiv$ source & $t \equiv$ sink

Def $f: V \times V \to \mathbb{R}$ is a <u>flow</u> for network G if

1) Capacity constraints: $f(u,v) \leq c(u,v)$
2) Skewed symmetric: $f(u,v) = -f(v,u)$
3) Flowin = Flowout for $u \in V - \{s,t\}$

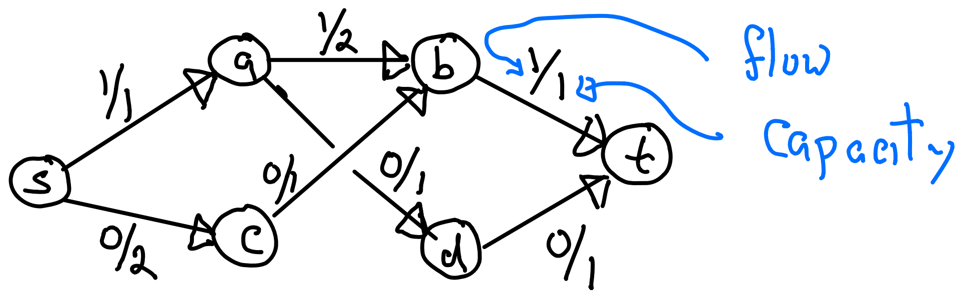
$$\sum_{v \in V} f(u,v) = 0$$

Def <u>Netflow</u> $\equiv |f| = \sum_{v \in V} f(s,v)$

Example:

Ex Network & flow

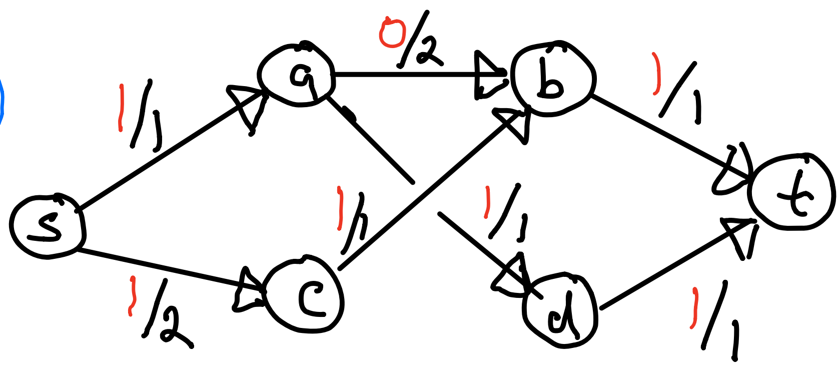
A)



Netflow = 1 (Greedy cannot improve)

Another flow:

B)



Netflow = 2 A) ⇒ B) required an undo

The Maximum Flow Prob

Input: Flow-Network $G=(V,E), s, t, c$

Output: Flow f with maximum net-flow.

Residual Network

Consider Network $G=(V,E,c)$ & flow f

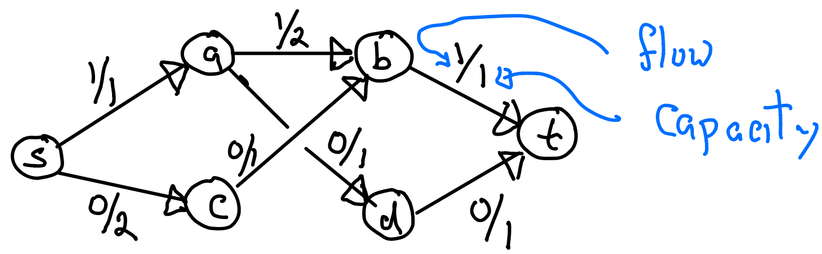
Def Residual Capacity: $c_f(u,v) = c(u,v) - f(u,v)$

Residual Network:

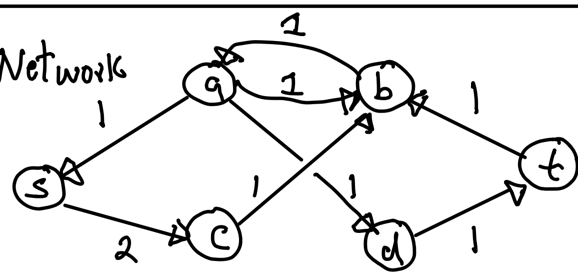
Edges $E_f = \{ (u,v) \in V^2 \mid c_f(u,v) > 0 \}$

$G_f = (V, E_f, c_f)$

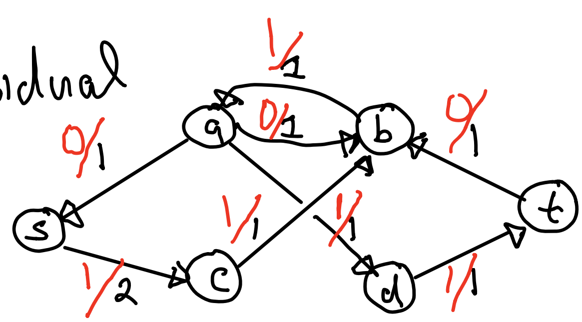
Ex Network & flow



Residual Network



Flow on residual



Ford-Fulkerson Method

F-F-Method (G, s, t, c)
1) Initialize flow f to zero.
2) While $\exists$ augmenting Path p in G_f
   Set f_p be max flow on p.
   Set $f = f_p + f$
3) Return f.

Def P is an aug path if P is a path in G_f using edges with positive capacities.

Lemma Let f be a flow on G & G_f its residual then

a) f' is a flow on G_f iff $f+f'$ is a flow on G.
b) f' is a max flow on G_f iff $f+f'$ is a max flow on G.
c) $|f+f'| = |f| + |f'|$ (f' has no flow in S)

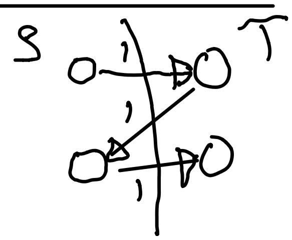
Pf a) $f'(e) \le c_f(e)$ iff $f'(e) \le c(e) - f(e)$
iff $f'(e) + f(e) \le c(e)$
iff $(f'+f)(e) \le c(e)$

S-T Cuts

Def $S, T \subseteq V$ is a cut if

1) $S \cap T = \emptyset$ & $S \cup T = V$
2) $s \in S$ & $t \in T$

Def $Cap(S,T) = \sum_{u\in S, v\in T} c(u,v)$



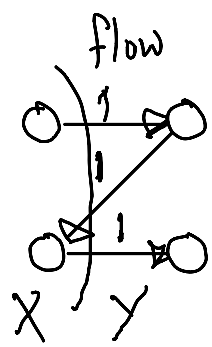
eg

$Cap(S) = 2$

Let f be a flow

net-flow:

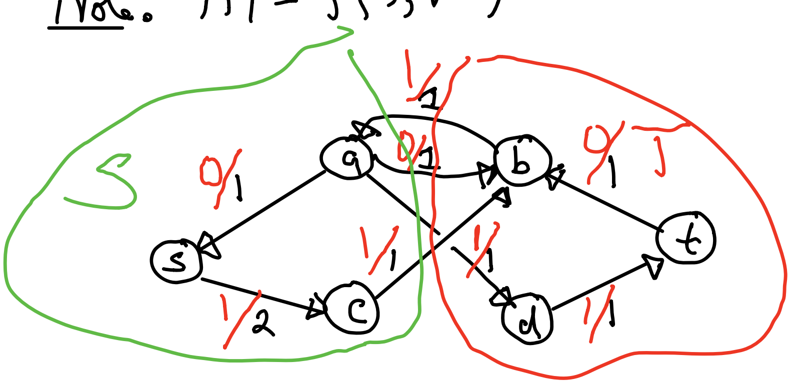
Def $f(X,Y) = \sum_{u\in X} \sum_{v\in Y} f(u,v)$



$f(X,Y) = 1$

Def: $f(S,T) \equiv$ net flow from S to T.

Note: $|f| = f(s, V-s)$



$Cap(S,T) = 3$

$f(S,S) = 1$

<u>Lemma</u> If f is a flow on G & (S,T) is a cut then $|f| = f(S,T)$.

<u>Pf</u> By induction on $|S|$

$|S|=1$ done by def.

Assume true sets of size less than S.

i.e. $|f| = f(S',T')$ where $S = S' \cup \{x\}$

Thus $S = S' \cup \{x\}$, $T = T' \setminus \{x\}$, $T' = T \cup \{x\}$

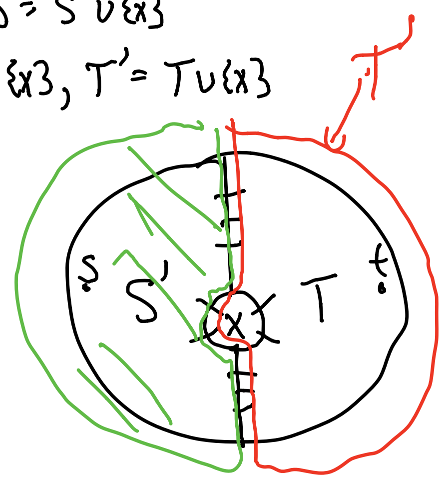
We get

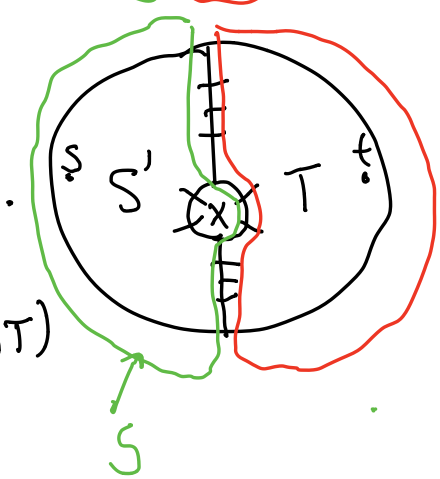
$$f(S',T') = f(S',T) + f(S',x)$$

$$f(S,T) = f(S',T) + f(x,T)$$

$$\begin{aligned} |f| &= f(S',T') \\ &= f(S',T) + f(x,T) \\ &= f(S,T) - f(x,T) + f(x,T) \\ &= f(S,T) \end{aligned}$$

flowin = flowout of x.





<u>Cor</u>: (S,T) cut then $|f| \le c(S,T)$

since $f(S,T) \le c(S,T)$

Thm Max-flow = Min-Cut

i.e. The following are equivalent

1) f is a max-flow
2) G_f contains <u>no</u> augmenting path.
3) $\exists$ cut (S,T) s.t. $|f| = Cap(S,T)$

We will show 1) $\Rightarrow$ 2) $\Rightarrow$ 3) $\Rightarrow$ 1)

1) $\Rightarrow$ 2) iff $\neg$2) $\Rightarrow$ $\neg$1) (contrapositive)

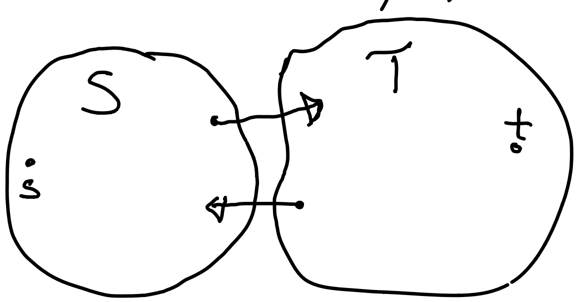
$\exists$ augmenting path $\Rightarrow$ f not maximum flow. (Lemma)

2) $\Rightarrow$ 3)

Let $S \subseteq V$ be nodes reachable from s in G_f.

Let $T = V - S$

Note: $t \in T$ by 2).



1) f saturates all edges in G from S to T.

2) f does not use edges from T to S in G.

$\therefore |f| = f(S,T) = Cap(S,T)$

3) $\Rightarrow$ 1) (Lemma)

Remove My Puppy from Image

Idea 1 Keep only white pixels!

1) No nose, mouth, eyes!

Idea 2 Also keep black pixels!

1) Some of puppy is neither!

Idea 3 View as a max-flow problem!

An Image Segmentation Problem

The Foreground-Background Problem

Input: 1) Pixel Image.

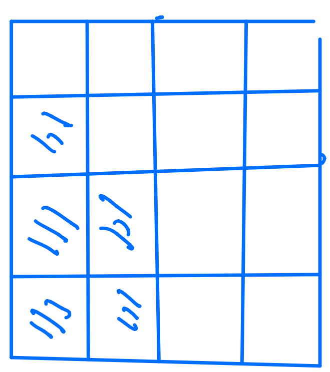
Graph: a) Pixels are vertices
b) Neighboring Pixels have an edge between.

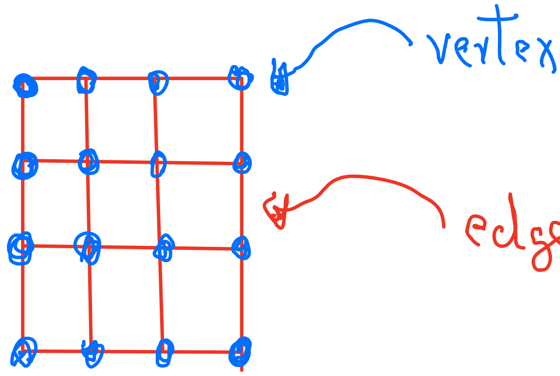
2) a_j is likelyhood V_j is in foreground.
3) b_j is likelyhood V_j is in background.

4) Weighted affinity graph.
weights $w_{ij} = w(v_i, v_j)$

---

eg


Pixel Image



Pixel Graph

e.g. One has picture of your puppy.

Goal: remove her from picture (pixels)

Suppose dog is white/black.

Thus

2) White/black pixels should be in foregrond. a's

3) Colored pixels in background b's

4) Neighboring white pixels should stay together

4) Neighboring black pixels should stay together

4) Neighboring colored pixels should stay together

edges w_{ij}

Write as an optimization prob.

Output: Partition A, B of V s.t.

$$\underset{A,B}{\text{Max}}\ q(A,B) \equiv \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{i \in A \\ j \in B}} w_{ij}$$

Change to Min Prob

Let $Q = \sum_i a_i + b_i$

Then

$$q(A,B) = Q - \sum_{i \in A} b_i - \sum_{j \in B} a_j - \sum_{\substack{i \in A}} w_{ij}$$

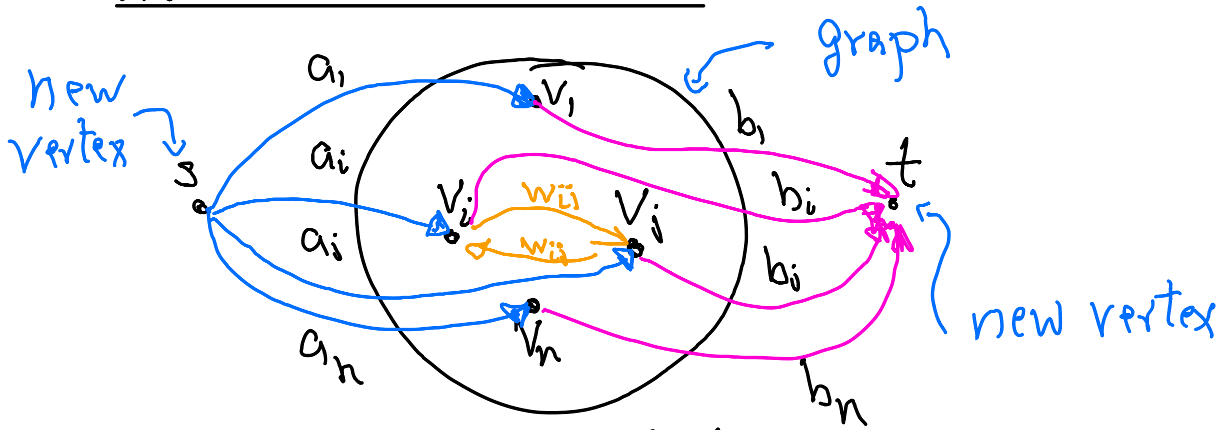
$j \in B$

Since Q is a constant

Suffice

$$\min q'(A,B) = \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{\substack{i \in A \\ j \in B}} w_{ij}$$

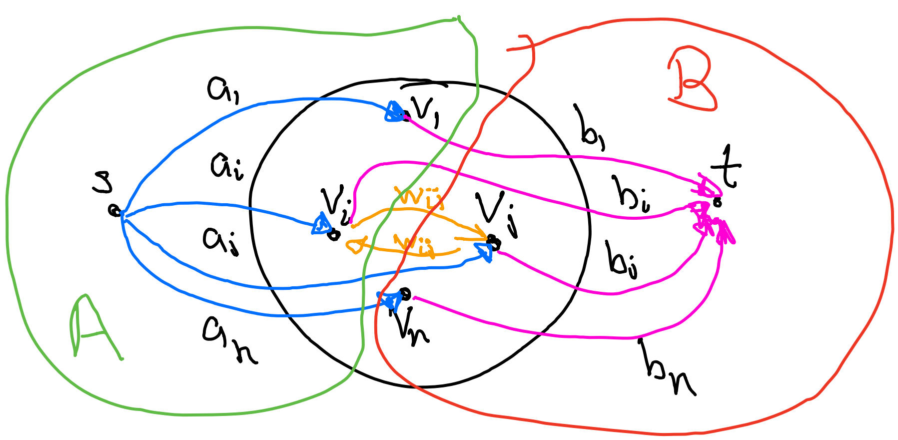
View as a flow network!



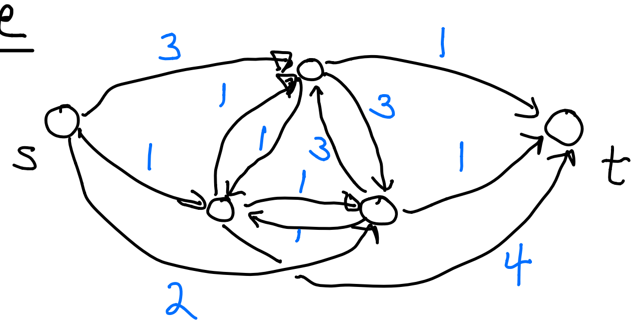
Claim If A, B is an S-T cut then

$$C(A,B) = q'(A,B)$$

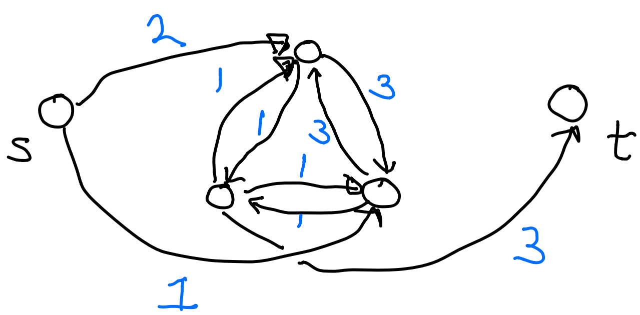
Thus Min Cut is Max Segmentation.



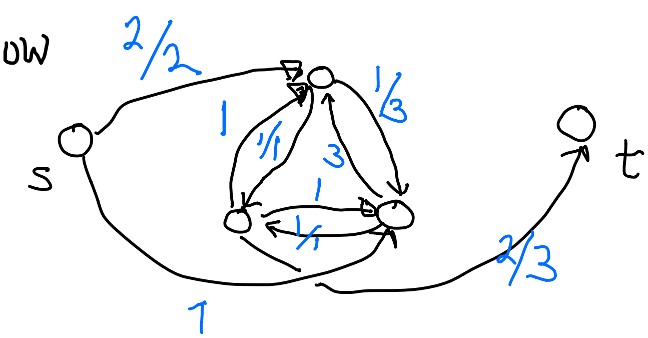
Example



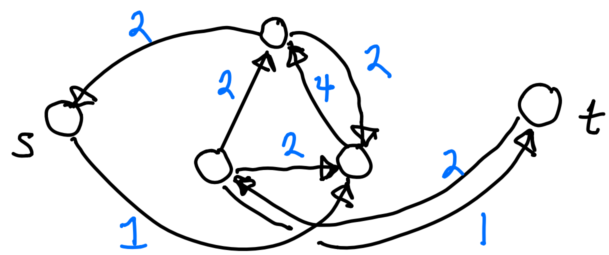
Note We may assume that $a_i = 0$ or $b_i = 0$



Consider flow f.

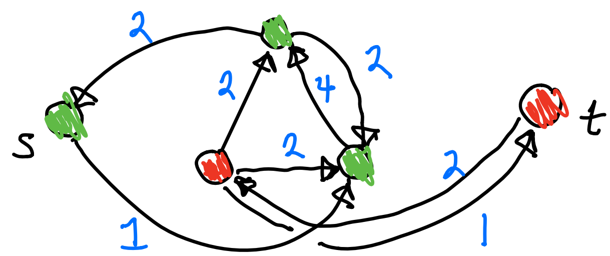


Residual

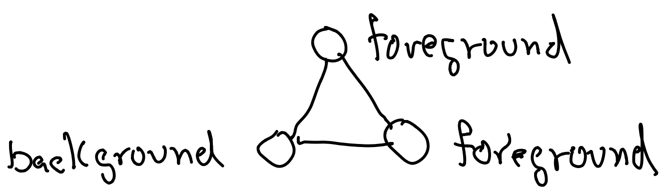


Reachable from S

Not reachable from S



Thus



Dynamic Network!

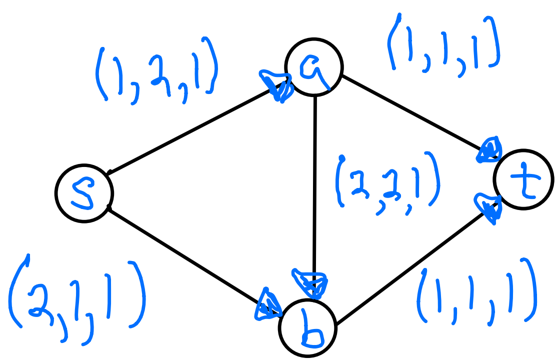
Input: Network $G = (V, E, s, t)$

Discrete Time $(0, 1, 2)$

Capacities c_{ij}^{t}

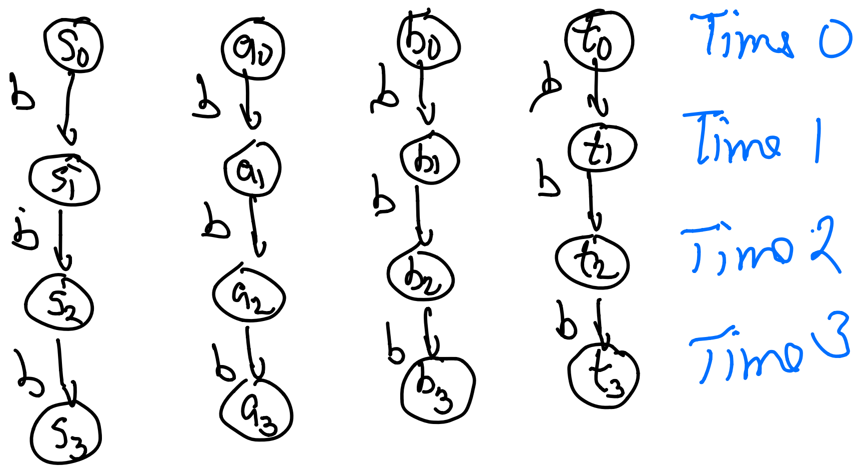
where $c_{ij}^{t} \equiv$ capacity from a_i to a_j at time t.

Buffer size $= b$.

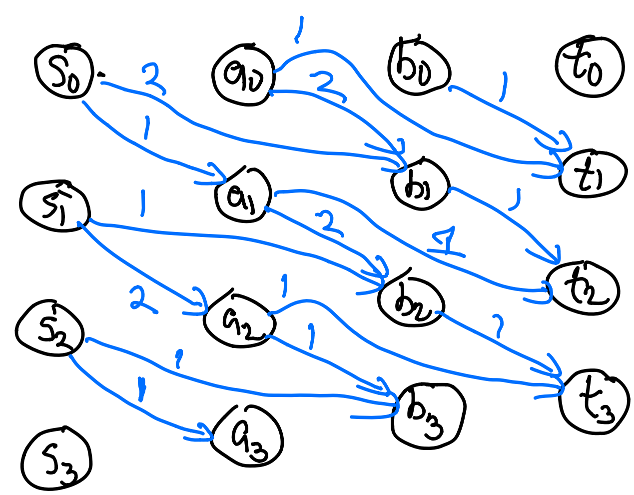


Idea: Make 4 copies of static network and add edges between copies.

Add buffer edges



Add transition edges.



Solve max flow from s_0 to t_3 on network of buffer & transition edges.