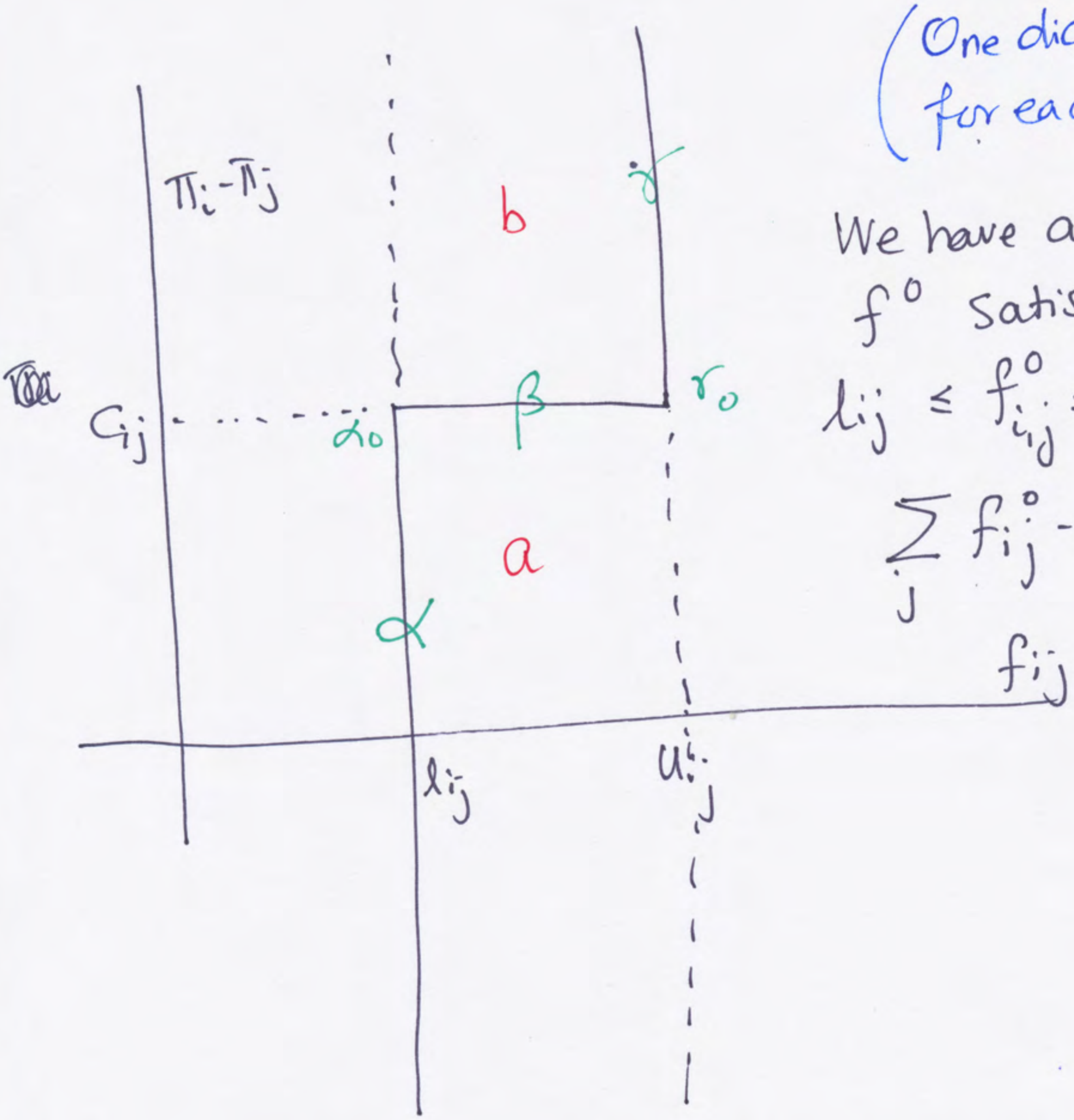
CS6381: Lecture #12 9.19.2020

Recall : ~~Out~~ C-S diagrams



(One diagram for each edge)

We have assumed that f^0 satisfies

$$l_{ij} \le f^0_{ij} \le u_{ij}$$

$$\sum_j f^0_{ij} - \sum_j f^0_{ji} = q_i \quad i \in V$$

Such f^0 can be found using max flow type algorithm with lower & upper bounds <u>and</u> external flows.

This is done only for Convenience; it is <u>NOT</u> required.

If all $(i,j) \in E$ are in states $\alpha, \alpha_0, \beta, \gamma, r_0$; f^0 is optimal

Our algorithm <u>progresses</u> towards making all edges to be in one of $\{\alpha, \alpha_0, \beta, \gamma, \gamma_0\}$ states. Starting from a solution that may not satisfy this condition. This algorithm is called (by FF) <u>Out-of-kilter</u> algorithm. States $\alpha, \alpha_0, \beta, \gamma, \gamma_0$ are said to be in-kilter <u>and</u> a, b are said to be ~~Out~~ <u>Out-of-kilter</u>. [The name comes from this]

Now we describe one possible version of this algorithm.

<u>Out-of-kilter algorithm:</u>

<u>Step 1</u>: Start with f^0 satisfying

$$l_{ij} \le f^0_{ij} \le u_{ij} \quad \forall (i,j) \in E$$

$$\sum_j f_{ij} - \sum_j f_{j,i} = q_i \quad i \in V$$

(~~using~~ using, if needed, a max-flow type algorithm. If no such f^0 exists, this algorithm will show that and ~~our~~ min-cost flow problem is infeasible and we stop)

Here after, we use two subroutines:

I. Flow change subroutine

II. Dual variable change Subroutine.

[Please note $x_{i,j}$, $y_{i,j}$ in (D) are implicit;
for any $\{\pi_i\}$, define $x_{ij} = \max(c_{ij} - (\pi_i - \pi_j), 0)$

$$y_{ij} = \max(\pi_i - \pi_j - c_{ij}, 0)$$

~~Exactly~~ at most one of the pair $\{x_{i,j}, y_{ij}\}$ can be positive.]

We describe these two subroutines and when we switch between the two. But the overriding idea is NEVER allow an edge in-kilter to go out-of-kilter; more edges that are out-of-kilter "closer" to in-kilter.

As such, flow is not permitted to change

a) for edges in state α or γ

b) for edges in State α_0, b, flow is only allowed to increase (within limits)

(c) for edges in state γ_0, a flow is only permitted to decrease (within limits)

(d) for edges in State β flow is permitted to either increase or decrease within limits.

During Dual Variable Changes:

(e) $(\pi_i - \pi_j)$ is allowed to increase without limit for edges in State γ, γ_0

(f) $(\pi_i - \pi_j)$ is allowed to decrease without limit for edges in State α, α_0

(g) $(\pi_i - \pi_j)$ is not allowed to increase for edges in State α_0, b

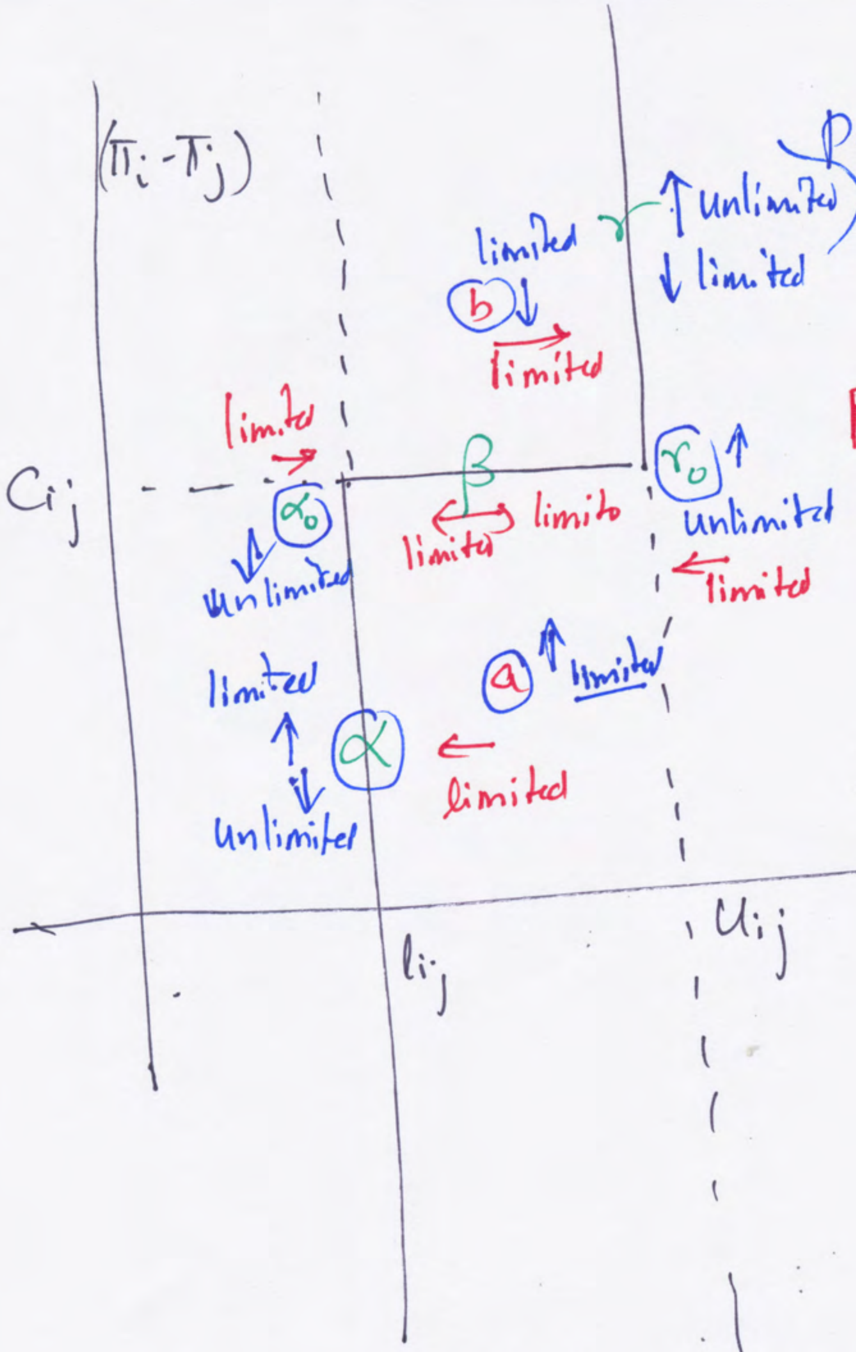
(h) $(\pi_i - \pi_j)$ is not allowed to decrease for edges in State γ_0, a

(i) $(\pi_i - \pi_j)$ is not allowed to change for edges in State β

(j) For edges in States $\{\gamma, b\}$ $(\pi_i - \pi_j)$ can decrease within limits

(k) For edges in States (α, a), $(\pi_i - \pi_j)$ can increase within limits.

These permitted changes are shown in the diagram on the next page

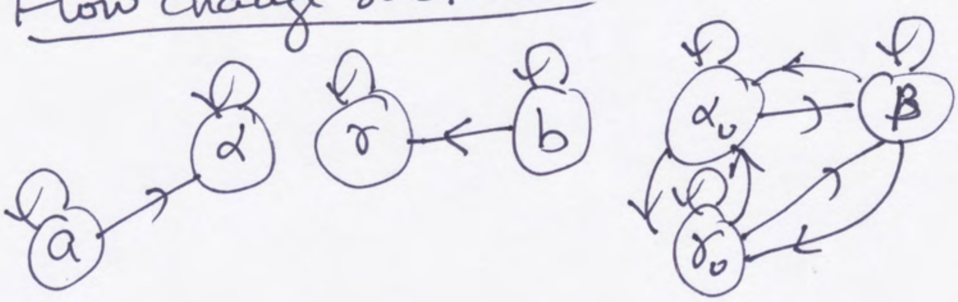


Possible flow change directions shown in ~~blue~~. red

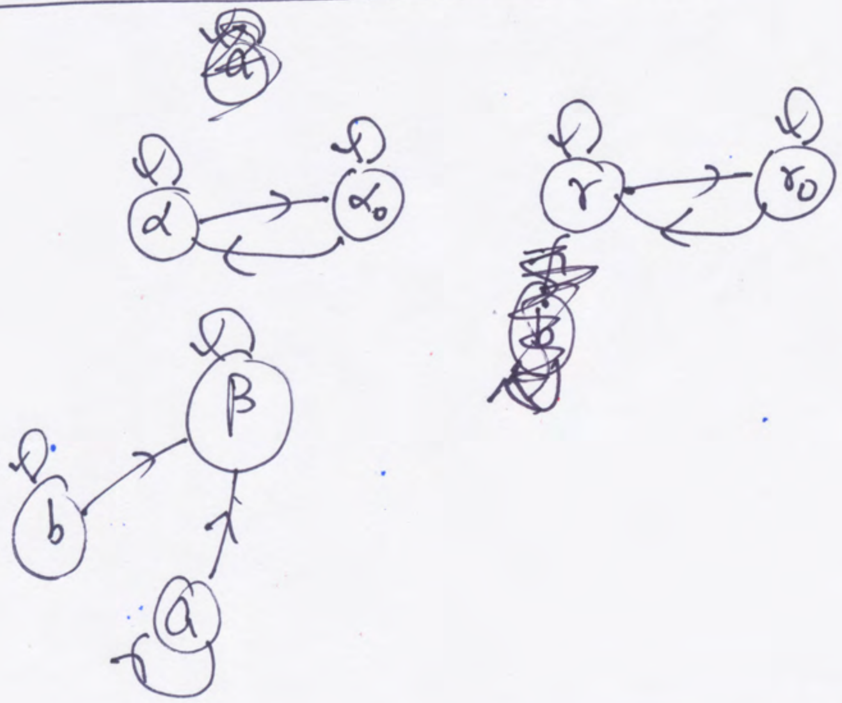
Possible dual change directions shown in ~~red~~. blue

Possible State transitions

Flow change subroutine



Dual Variable Change Subroutine



Whenever an edge goes from $\{a, b\}$ to $\{\alpha_0, \alpha, \beta, \gamma_0, \gamma\}$ we decrease the number of edges that are out-of-kilter.
Else we decrease another measure called Kilter-number that shows we are making progress towards optimality.

Kilter-Number can be defined several ways and we get appropriate versions of the algorithm. We define these in the next page.

The one used in FF-book: first.

For edges in states $\{\alpha, \alpha_0, \beta, \gamma, \gamma_0\}$ Kilter-number is 0. If $\sum_{ij \in E} k_{ij} = 0$, we have optimality.

For edge ij in state a: $[f_{ij} - l_{ij}] \cdot [c_{ij} - (\pi_i - \pi_j)]$

b: $[u_{ij} - f_{ij}] \cdot [(\pi_i - \pi_j) - c_{ij}]$

In either case kilter number is positive for edges in state (a) or (b). Hence if $\sum k_{ij} = 0$, no edge is out-of-kilter & hence optimality.

This is our goal and we never let $\sum_{ij \in E} k_{ij}$ <u>increase</u> in our algorithm. We decrease it at each step.

Now describe our algorithm. It starts with the selection of an edge in state a or b and attempts to bring it in-kilter while not allowing any edge to go out-of-kilter (Even other edges may decrease k_{ij} in the process)

Suppose we have a feasible flow f^0

i.e $l_{ij} \le f^0_{ij} \le u_{ij} \quad \forall (i,j) \in E$

$$\sum_j f^0_{ij} - \sum f^0_{j,i} = q_i \quad \forall i \in V$$

Let $\{\pi^0_i\}$ be chosen arbitrarily.

For this (f^0, π^0), we can find the state of each edge. If none are in $\{a^*, b\}$, then we are done. f^0 is opt for P

π^0 is opt for D

(with suitably defined x^0, y^0)

So suppose not. Let edge (p,q) be in State a without loss. (Similar methods with suitable modification apply if (p,q) is in State ~~a~~ b)

<u>Flow changes</u> : Desirable to <u>decrease</u> (increase) flow on (p,q) if it is in State a (b)

But ~~to~~ to do this, we need to increase flow on some <u>admissible</u> path from p to q.

Consider a "residual" graph $G^{(f^0;\pi^0)}$ as follows:

$$(i,j) \in E^{(f^0;\pi^0)} \Leftrightarrow (i,j) \in E;\ \text{state: } \alpha_0, \beta, b$$

(OR)

$$(j,i) \in E;\ \text{State } \gamma_0, \beta, a$$

residual capacities: in dir $i \to j$ $\left\{ \begin{array}{l} \text{for } (i,j) \in E;\ \text{State } \alpha_0, \beta, b \\ \quad u_{i,j} - f^0_{i,j} > 0 \\ \text{for } (j,i) \in E;\ \text{State } \gamma_0, \beta, a \\ \quad f^0_{ij} - l_{ij} > 0 \end{array} \right.$

Origin p, destination q, in state a

Send min $\left\{ \begin{array}{l} \text{max flow in } G^{(f^0,\pi^0)} \text{ from } p \text{ to } q; \\ \cdot\ f^0_{p,q} - l_{p,q} > 0 \end{array} \right.$

If the above min $= f^0_{p,q} - l_{p,q}$; $(p,q) \to$ State α

One more edge in Kilter.

If not, $\exists$ a min-cut in $G^{(f^0,\pi^0)}$ separating p and q

$\downarrow$

$(S, \bar{S})$

$$\pi_i^{new} = \begin{cases} \pi_i^{old} + \delta & i \in S \\ \pi_i^{old} & i \in \bar{S} \end{cases}$$ where δ is given by (next page

Let $\delta_1 = \min\limits_{\substack{(i,j):\gamma, b \\ i\in\bar{S}, j\in S}} \{\pi_i - \pi_j - c_{ij}\} > 0$

$\delta_2 = \min\limits_{\substack{\alpha\ a \\ i\in S \\ j\in\bar{S}}} \{c_{ij} - (\pi_i - \pi_j)\} > 0$

$\delta = \text{Min}(\delta_1, \delta_2)$

[At max flow $< (f^0_{p,q} - l_{p,q})$,

$i \in S$: ~~Reachab~~ reachable from p

$i \in \bar{S}$: not " "

$p \in S$, $q \in \bar{S}$

~~i, j~~ $i \in S, j \in \bar{S}$: $\Rightarrow$ (i,j) in states: γ, γ_0, a, ~~α~~

$i \in \bar{S}, j \in S$: $\Rightarrow$ (i,j) " " $\alpha_0, \alpha, b, \gamma$

$$\pi_i^{new} - \pi_j^{new} = \begin{cases} \pi_i^{old} - \pi_j^{old} & i\in S, j\in S \\ & i\in\bar{S}, j\in\bar{S} \\ \pi_i^{old} - \pi_j^{old} + \delta \uparrow & i\in S, j\in\bar{S} : \underbrace{\gamma, \gamma_0}_{\text{unlimited}} \quad \underbrace{\alpha, a}_{\text{limits}} \\ \pi_i^{old} - \pi_j^{old} - \delta \downarrow & i\in\bar{S}, j\in S : \underbrace{\alpha_0, \alpha}_{\text{unlimited}} \quad \underbrace{b, \gamma}_{\text{limits}} \end{cases}$$

After each dual variable change, we go back to flow changes (possibly many aug. paths)

$\rightarrow$ Increases size of BFS-tree at each step if no flow increase results.

An Example :

$l_{ij} = 0 \;\; \forall (i,j) \in E$

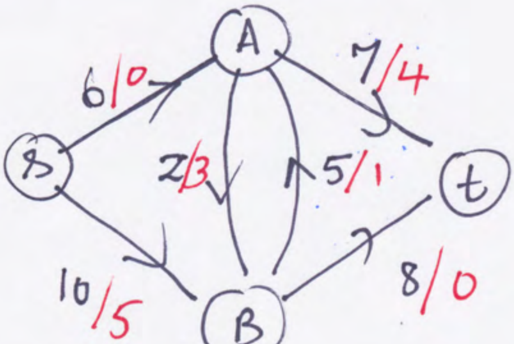
u_{ij} : black.

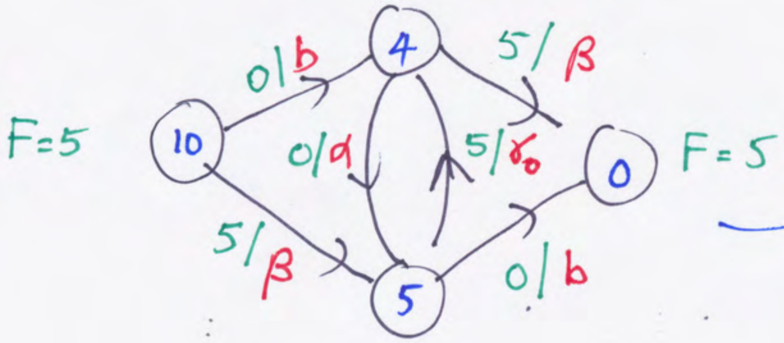
f_{ij} : green

c_{ij} : Cost

π_i : blue

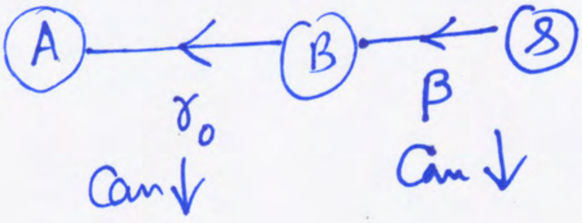
State : red





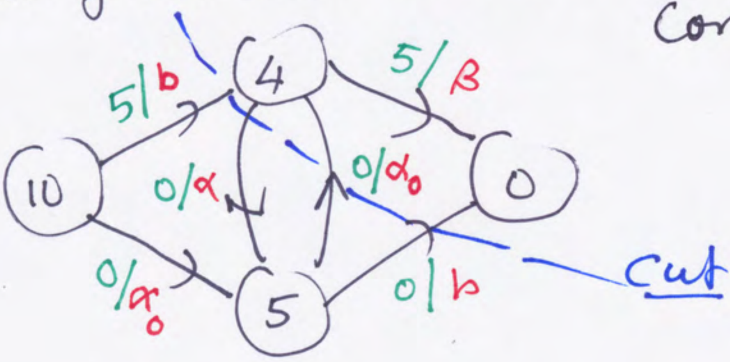
Select edge (s, A) in state b : Want to ↑ flow on (s, A)

So find an admissible path from A to s



So decrease by 5 units on (s, B), (B, A) and increase on (s, A) by 5 units.

New flows and states



Continue with (s, A) in state b.

$\delta_1 =$

$\delta_2 =$

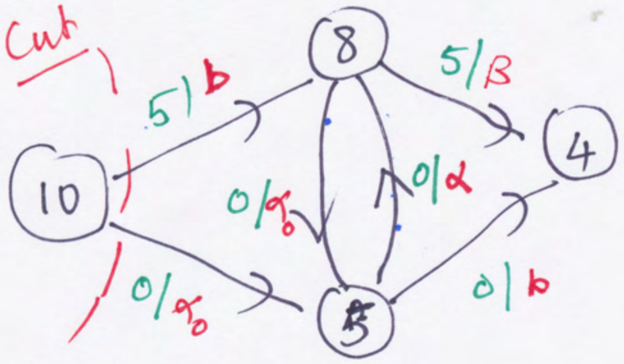
Dual Variable Change

$$S = \{A, t\}, \quad \bar{S} = \{s, B\}$$

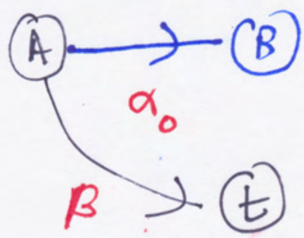
$$\delta_1 = \text{Min}\left[\underset{\substack{(s,A)\\ \text{State } b}}{10-4-0},\ \underset{\substack{(B,t)\\ \text{State } b}}{5-0-0}\right] = 5$$

$$\delta_2 = \text{Min}\left[\underset{\substack{(A,B)\\ \text{state } \alpha}}{3-(4-5)}\right] = 4$$

$$\delta = \text{Min}(\delta_1, \delta_2) = 4.$$



Continue with (s, A) in State b:
So try to find an admissible path from A to s

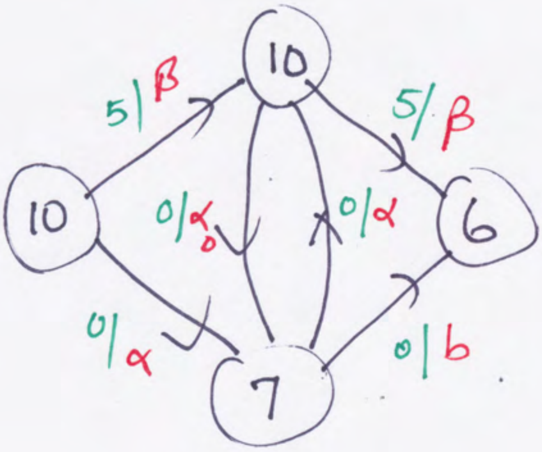


$$S = \{A, B, t\} \quad \bar{S} = \{s\}$$

$$\delta_1 = \text{Min}\left[\underset{\substack{(s,A) \text{ in}\\ \text{State } b}}{10-8-0}\right] = 2$$

$$\delta_2 = \infty \qquad \delta = 2:$$

Dual Variable change.

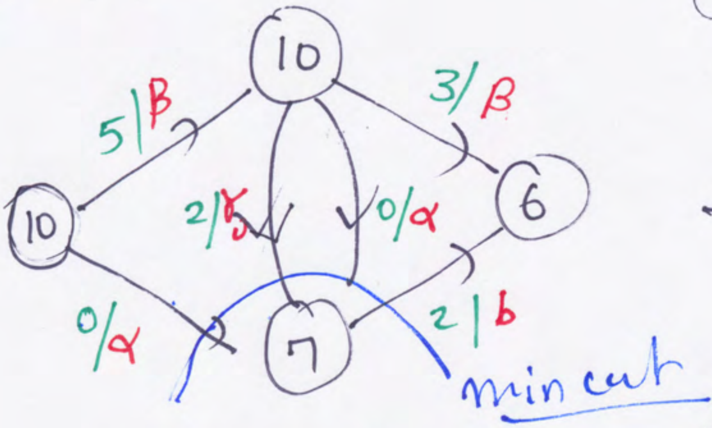


edge (s, A) now in-kilter

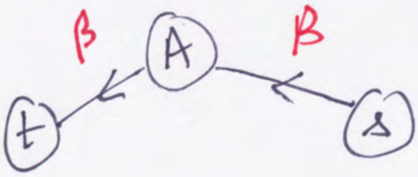
So we next take up edge (B, t) in state b. Again wants to ↑ flow on (B, t). So try to find an admissible path from t to B ~~to~~.

t - A - B : 2 (limited by A, B)

New flows:



Continue to try to ↑ flow on (B, t) in State b.



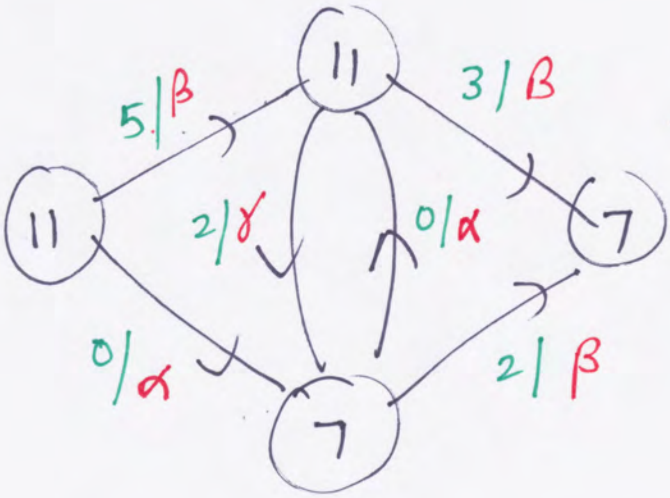
$S = \{s, A, t\} \quad \{B\} = \bar{S}$

$\delta_1 = \text{Min}\left[\underset{\substack{(B,t)\\ \text{in state } b}}{7-6-0}\right] = 1$

$\delta_2 = \text{Min}\left[10-7-1\right] = 2$

$\delta = 1$

Dual Variable Change



Optimal Solution to (P) and (D)

End

Each of Flow subroutines are strongly polynomial.

\# of dual variable changes without changing flows is also $\leq |V|$. Since BFS tree enlarges at each step. The only issue is that the number of alternative cycles of flow-change - dual variable change - flow change . . seems <u>not</u> to have an appropriate bound.

To rectify this we use "Scaling" techniques (very much like "bit" operations in CS). This is due to J. Edmonds & R.M. Karp. And we take this up next.