In order to improve on Dinic's algorithm, we try to "speed" up the process of Saturating LN.

Recall: LN is an acyclic directed graph. All edges go from a node in "lower" layer to a "the next higher" layer.

What we need for this step, is that it (LN) is a DAG whose topolgical order is known. and edges go from ith to (i+1)st layer

Instead of saturating a path each time (which Saturates one edge of the path) we now try to saturate a node at a time. This is known as MKM (Maheswari, Kumar, Malhotra) method.
S.N M. Pramodh V, M

(Information Proc. Letters Oct, 1978 Vo7 # 6 p 277-278)

We calculate what are known as node-potentials as follows:

For each node $v \neq s, t$,

$$\rho_{out}(v) = \sum_{(v,x)\in LN} c^f(v,x) \; ; \quad \rho_{in}(v) = \sum_{(x,v)\in LN} c^f(x,v)$$

$$\rho(v) = \min\left[\rho_{out}(v), \rho_{in}(v)\right]$$

$$\rho(s) = \rho_{out}(s) = \sum_{(s,x)\in LN} c^f(s,x) \quad ; \quad \underbrace{\rho_{in}(t)}_{\rho(t)} = \sum_{(x,t)\in LN} c^f(x,t)$$

$$\rho = \min_{v\in LN} \rho(v) = \rho(r)$$

Node potential tells us the maximum that can pass through that node

There are three possibilities 1) $r = s$, 2) $r = t$, 3) $r \neq s, t$.

Third is most general. We will show how to Saturate r and remove it from LN in the Current phase. So each phase has atmost $|V|$ subphases. Visualize LN (remaining) in this sub phase as follows

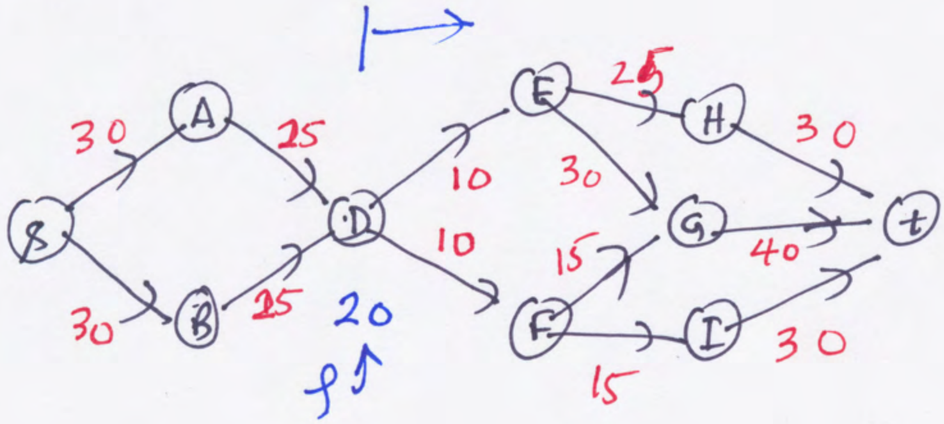
Remove temporarily all other nodes in the layer of r

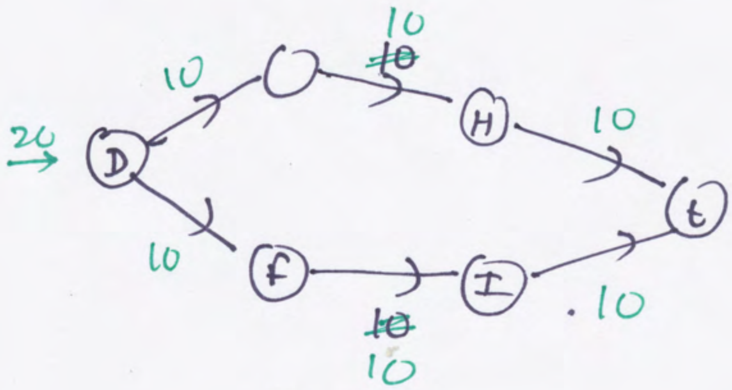
all nodes prior to r in top. order (layer#)

all nodes after r in top. order layer

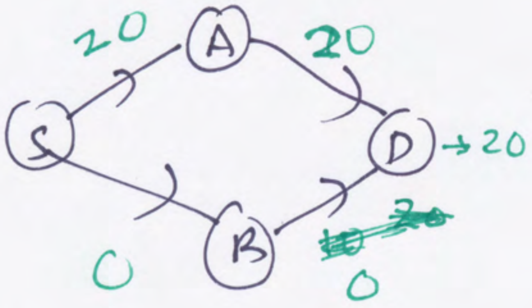
$\longleftarrow$ pull $\qquad$ $\longrightarrow$ push

Example





Push



Each time we send flows, we reduce residual Capacities of edges and ADJUST node potentials

Remove nodes with 0 potential and their edges

Remove edges with 0 residual capacities.

This saturates a node in $\Theta(|V|)$ time

Thus, with proper implementation, MKM solves the problem in $O(|V|^3)$ Steps.

The other way of achieving this is due to A.V. Karzanov. Called Preflow-Push-Relabel algorithm. This is done in CLRS and my notes on my home page follow CLRS description

Maybe one group can present this.

Lower Bounds and Exogeneous Flows

Now we discuss two related extensions that are not as "easy" as taking care of (i) Undirected edges (ii) Node capacities and (iii) Multiple Origins/destinations (Single "commodity")
The "latter" were covered in CS 6363 (I hope!)

Lower Bounds

Input: For each edge, (l_{ij}, u_{ij}) with $l_{ij} \le u_{ij}$ are given:

l_{ij} : lower bound on flow f_{ij}

u_{ij} : upper bound " f_{ij}

So the problem becomes

$$l_{ij} \le f_{i,j} \le u_{ij} \qquad \forall\ (i,j) \in E$$

$$\sum_j f_{i,j} - \sum_j f_{j,i} = \begin{cases} F & i = s \\ 0 & i \ne s,t \\ -F & i = t \end{cases}$$

Maximize F.

If we have a feasible flow f^0, we can do the algorithm with small changes.

$$c^f(i,j) = \underbrace{(u_{ij} - f^0_{ij})}_{\ge 0} + \underbrace{(f^0_{j,i} - l_{ji})}_{\ge 0} \ge 0 \qquad \forall\ i \in V,\ j \in V,\ j \ne i$$

$$E^f = \{ (i,j) : c^f(i,j) > 0 \}$$

Rest of the algorithm is the same. If u_{ij}, l_{ij} are integral and f^0_{ij} is integral, integrality is maintained at all steps.

The most important question: How to get f^0?

Is it possible that no such f^0 exists? How to check this?

This question did NOT arise when $l_{ij} = 0 \ \forall i, j$ since $f \equiv 0$ was a feasible solution!.

Exogeneous flows

Here, there may be incoming/outgoing net flows at each node and they are part of the input. We let q_i be the "net" out flow at node i (part of the data). We may also have lower and upper bounds in general. So the problem look like

$$l_{i,j} \le f_{i,j} \le u_{ij} \qquad \forall (i,j) \in E$$

$$\sum_j f_{i,j} - \sum_j f_{j,i} = \begin{cases} q_s + F & i = s \\ q_i & i \ne s, t \\ q_t - F & i = t \end{cases}$$

$$\text{Max } F$$

Data: $[\, l_{ij},\ u_{ij},\ q_i \,]$

(7)

However, we can easily convert this to the case where $l_{ij} = 0 \ \forall ij$

Define $\tilde{f}_{ij} = f_{i,j} - l_{ij} \quad \forall (i,j) \in E$.

Let $\tilde{q}_i = \sum_j l_{ij} - \sum_j l_{j,i} \quad \forall i \in V$

Problem changes to

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

$$\sum_j \tilde{f}_{i,j} - \sum_j \tilde{f}_{j,i} = \begin{cases} q_s + \tilde{q}_s + F & i = s \\ q_i + \tilde{q}_i & i \ne s,t \\ q_t + \tilde{q}_t - F & i = t \end{cases}$$

Max F.

So, it is generally assumed that lower bounds are 0 for the Exogeneous Flow Problem

Each of {Lower Bound Problem, Exogeneous Flow Problem} can be converted to the other.

Lower Bound → Exogeneous Flow

Let $\tilde{f}_{i,j} = f_{i,j} - l_{ij}$

So the problem becomes

$$0 \le \tilde{f}_{ij} \le u_{ij} - l_{ij}$$

$$\sum_j \tilde{f}_{i,j} - \sum_j \tilde{f}_{j,i} = \begin{cases} \tilde{q}_s + F & i = s \\ \tilde{q}_i & i \ne s, t \\ \tilde{q}_t + F & i = t \end{cases}$$

Max F

Converse: Left to you as exercise

Hence we only consider Lower Bound Case from now on. Again, our focus is to find an initial feasible Solution if one exists or to show one does NOT exist if that is the case in "polynomial" time.

We show this "itself" is also a maximum flow problem in which finding an initial feasible solution is "easy".

Let the original problem be:

$G = [V; E]$, $s \in V$, $t \in V$, $s \neq t$

$\{l_{ij}\}$ $\{u_{ij}\}$ for $(i,j) \in E$ such that

$l_{ij} \leq u_{ij}$ $\forall$ $(i,j) \in E$.

[Note that if is Not true for some (i,j) Problem is <u>infeasible</u>]

We create "another" problem.

$$\tilde{G}^{\#} = \{ [\underbrace{V \cup \{\tilde{s}, \tilde{t}\}}_{\tilde{V}}], \tilde{E} \}$$

$\tilde{s}$ is "new" origin, $\tilde{t}$ is new destination

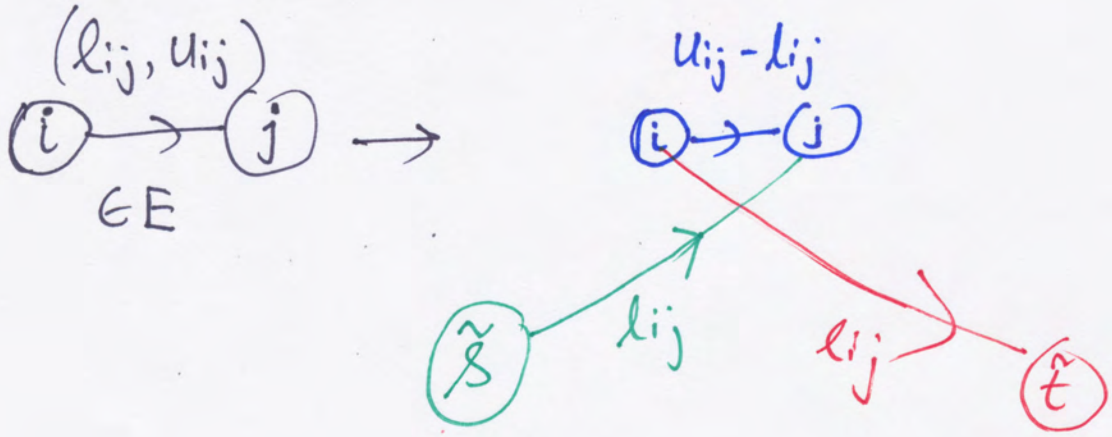
$\tilde{E}$ requires a bit more explanation.

$(t, s) \in \tilde{E}$; $l_{t,s} = -\infty$; $u_{t,s} = +\infty$

(this will not cause problem; this <u>is</u> <u>the</u> only edge in $\tilde{E}$ that has a "lower" bound. More edges to come.

An edge in E gives rise to 3 edges in $\tilde{E}$ as follows:

Edges in E are in black.



all these edges have 0 lower bound.

$\therefore |\tilde{V}| = |V| + 2$

$|\tilde{E}| = 3|E| + 1$

In $\tilde{G}$, flow $\equiv 0$ is feasible.

We want to maximize flow from $\tilde{s}$ to $\tilde{t}$ in $\tilde{G}$. There are two possibilities

(i) $\tilde{F}^* = \sum_{(i,j)\in E} l_{ij}$; (ii) $F^* < \sum_{(i,j)\in E} l_{ij}$

Original prob: feasible, Orig. prob infeasible.

Proof: If $\tilde{F}^* = \sum_{(i,j)\in E} l_{ij}$ then ~~g~~ red and green edges are <u>full</u>; let $\{\tilde{f}^*_{ij}\}$ be the corresponding edge flows in $\tilde{G}$.

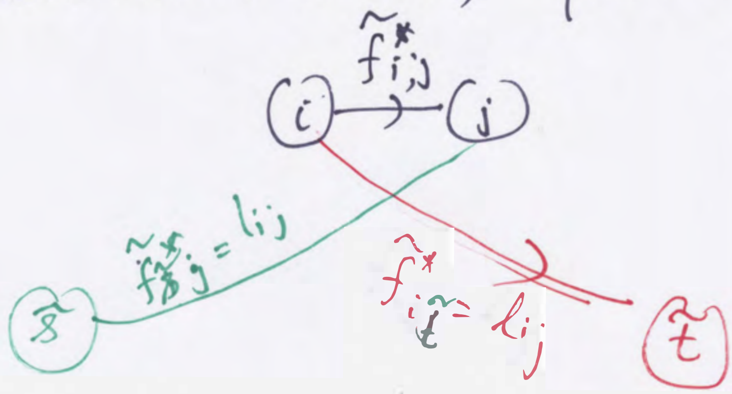
∵ Since for each black edge in $\tilde{G}$,

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

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

Let $f_{ij} := \tilde{f}^*_{ij} + l_{ij}$ $\quad$ (**)

∵ At each node i in $\tilde{G}$, for the three edges



Out going from i: $\tilde{f}^*_{ij} + l_{ij} = f_{ij}$ in G

incoming to j: $\tilde{f}^*_{ij} + l_{ij} = f_{ij}$ in G.

∴ Conservation is maintained.

∴ $\{f_{ij}\}$ defined in (**) is feasible to G.

Conversely, if problem on G is feasible, let $\{f^0_{ij}\}$ be such a feasible flow

Define $\tilde{f}^0_{ij} = f^0_{ij} - l_{ij} \quad (i,j) \in E$

$\tilde{f}_{s,j}$ on edge "arising" from $(i,j) \in E$

$= l_{ij}$

$\tilde{f}_{i,t}$ on edge "arising" from $(i,j) \in E$

$= l_{ij}$

Exercise : This set of values is feasible for $\tilde{G}$ problem and $\tilde{F} = \sum_{(i,j)\in E} l_{ij}$

Hence max flow in $\tilde{G}$ is $\sum_{(i,j)\in E} l_{ij}$ (it can't be more since cut $\{\tilde{s}\}\,\{\tilde{V}-\tilde{s}\}$ has this capacity.

Hence the problem of finding initial feasible flow can be done in $O(|V|^3)$ time.

Moreover, if $\{l_{ij}\}$, $\{u_{ij}\}$ are integral, this flow is also integral. Hence this problem has integrality property