Lecture 17 and Lecture 18

$$C = [C_{ij}]_{n\times n}$$

$X_{ij} = 1$ Perm I is assigned task j

$$\sum_i^1 x_{ij} = 1$$

$$\sum_j^1 x_{ij} = 1$$

Minimize

$$\sum_i^1 1 \sum_j^1 c_{ij X_{ij}}$$

$$C^1 = \begin{array}{c} V_1 \\ V_2 \\ V_3 \\ V_n \end{array} \begin{bmatrix} u_1 & u_2 & u_3 & u_n \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} n \times n$$

$$C_{ij} - (U_i + V_j) \geq 0 \begin{cases} C^1_{1j} = C_{ij} - (U_i + V_j) \\ C^1_{1j} \geq 0 \end{cases}$$

$$C_{ij} \geq (U_i + V_j)$$

$$\text{Min } \sum_i^1 1 \sum_j^1 c_{ij X_{ij}}$$

$$\sum_i^1 X_{ij} = 1$$

$$\sum_i^1 X_{ij} = 0$$

$$X_{ij} \geq 0$$

$$\sum_i^1 X_{ij} = 1 \quad \mathrm{U_i} \quad \forall i = 1,,,,,,,,,n$$

$$\sum_i^1 X_{ij} = 1 \quad \mathrm{V_i} \quad \forall i = 1,,,,,,,,,n$$

$$(u_{1+}v_1)\, x_{11} + (u_{2+}v_2)\, x_{22} + (u_{1+}v_2)\, x_{12} + (u_{2+}v_1)\, x_{21} + \ldots\ldots\ldots + (u_{n+}v_n)\, x_{nn}$$

$$= u_{1+}u_{2+}\ldots\ldots\ldots\ldots u_{n+}V_{1+}V_{2+\ldots\ldots\ldots\ldots}V_n$$

$$\sum_i^1 1 \sum_j^1 (u_{i+}v_j)\, x_{ij} = \sum_i^1 U_i + \sum_i^1 V_j$$

$$\sum_i^1 U_i + \sum_i^1 V_j = \sum_i^1 1 \sum_j^1 (u_{i+} v_j)\, x_{ij} \leq \sum_i^1 1 \sum_j^1 c_{ijX_{ij}}$$

$$\sum_i^1 U_i + \sum_i^1 V_j \leq \sum_i^1 1 \sum_j^1 c_{ijX_{ij}}$$

Dual optimization

$$Max \sum_i^1 U_i + \sum_i^1 V_j$$

$$Subject\ to\ C_{ij} \geq (U_i + V_j)$$

Min cost flow network problem
Minimize $\sum_i^1 1 \sum_j^1 c_{ijX_{ij}}$
St $x_{12}+ x_{13}= b_1 x w_1$
- $x_{12}+ x_{24} + x_{25}= b_2 x w_2$
- $x_{13}+ x_{34} + x_{35}= b_3 x w_3$
- $x_{24}- x_{34} + x_{45}= b_4 x w_4$
- $x_{25}- x_{45} - x_{35}= b_5 x w_5$

(w_1 - w_2) X_{12} + (w_1 - w_3) X_{13} + (w_2 - w_4) X_{24} +.........+ = $\sum_j^1 W_{ib_2}$

$$\sum_j^1 (W_{i-} W_j) X_{ij} = \sum_j W_{ib_1}$$

If we apply following constraint

$$W_{i-} W_i \leq C_{ij}$$

$$\sum_j^1 W_{ib_1} = \sum_i^1 1 \sum_j^1 (W_{i-} W_j) X_{ij} \leq \sum^1 c_{ijX_{ij}}$$

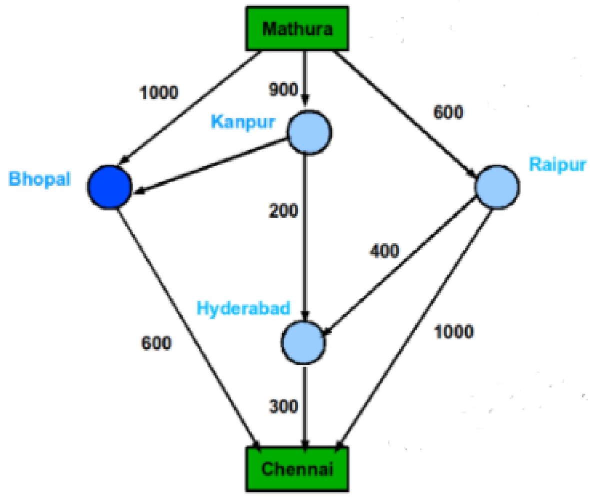
Lecture 18
This lecture centres around the below topics

- *Motivation behind Maximum Flow problem*
- *Maximum Flow Problem*
- *Max-Flow min-cut Theorem*
- *Ford Flukerson Algorithm*
- *Push relabel algorithm*

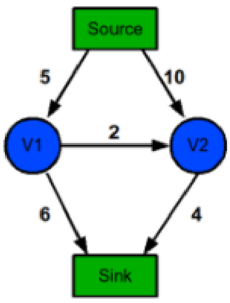
Maximum flow
The below graph implicates the oil refinery at Mathura producing oil and it has a warehouse in Chennai with each fluid flow. The goal is to identify which following model has to be used?

- *Liquids following through pipes*
- *Current through electrical networks*

- *Information through communication networks*
- *Vehicles through roads*

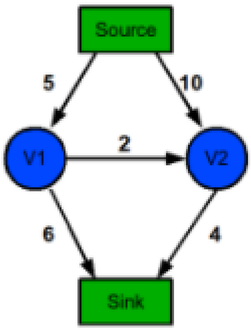


As analysing the flow network, each vertex other than source and sink is a conduit Junction which doesn't store/collect any material. 2 Each edge can be thought of a conduit for the material with a predefined capacity. Now we have to analyse what will be the maximum flow of the network.
As per the below graph
$G = (V,E)$
Where
$(u, v) \in E$

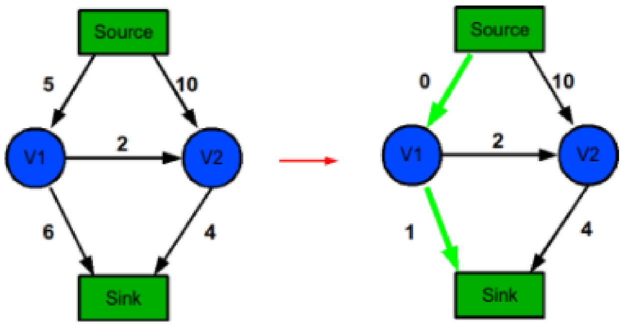


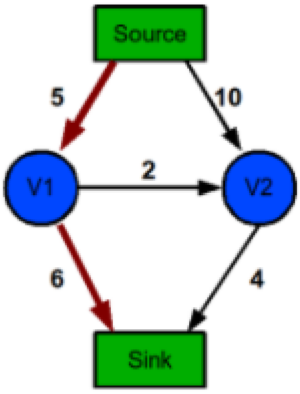
Let the flow of the network be defined as
$G = (V,E)$
Then a flow in
G is defined as a real valued function $f: V \times V \longrightarrow R$
And it should satisfy the below properties
Capacity Constraint
$f(u, v) \leq c(u, v); \forall u, v \in V$
Skew Symmetry

$f(u,\ v) = -f(v,\ u);\ \forall u,\ v \in\ V$
Flow Conservation
$\sum_{u \in V} f(u,v) = 0;\ \forall u \in\ V - \{s,\ t\}$
Total net flow is
Total positive flow entering a vertex $v = \sum_{u \in V} f(u,v) > 0 f(u,v)$, *which similarly how the total positive flow leaving a vertex is defined.*

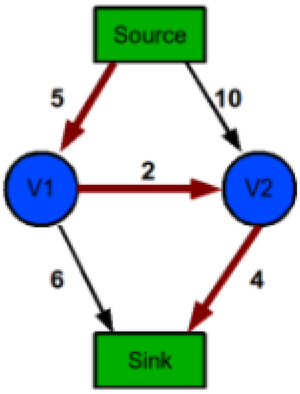


Maximum flow
Residual network
$Gf = (V,Ef)$
where
$Ef = \{(u,\ v) \in\ V \times\ V : Cf(u,\ v) > 0\}$
Residual capacity
The amount of flow we can push from u to v before exceeding the capacity c(u,v) is the residual capacity of c(u,v).





Residual capacity of path: is the minimum residual capacity along the path, where p is a simple from s to t, which is 5



Residual capacity of path: is the minimum residual capacity along the path, where p is a simple from s to t, which is 5.

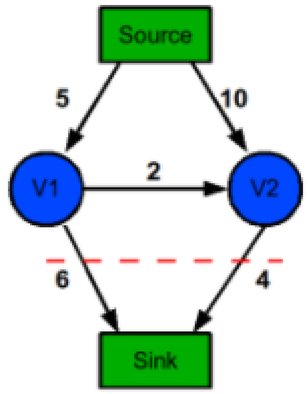
Cut

A cut C(S, T) of flow network G = (V,E) is a partition of set of vertices V into two sets disjoint sets S and T. Capacity of a cut is the capacity of edges going from vertices belonging to S to vertices belonging to set T.

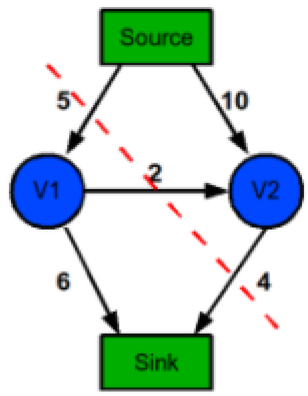
Cut capacity = 10

Cut capacity = 10

Cut capacity = 10



Cut capacity = 10

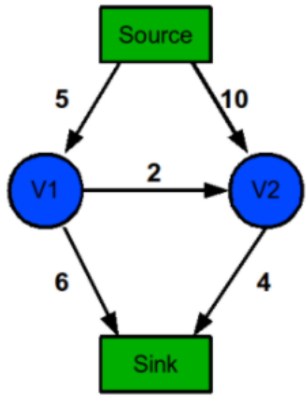


Cut capacity = 9

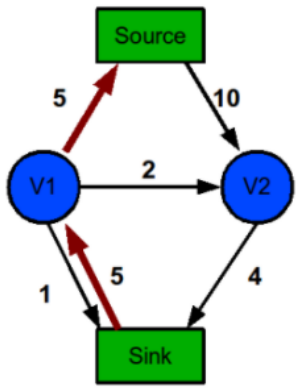
Max flow – min cut theorem

If f is a flow in flow network G = (V, E) with sources s and sink t, then the following conditions are equivalent:

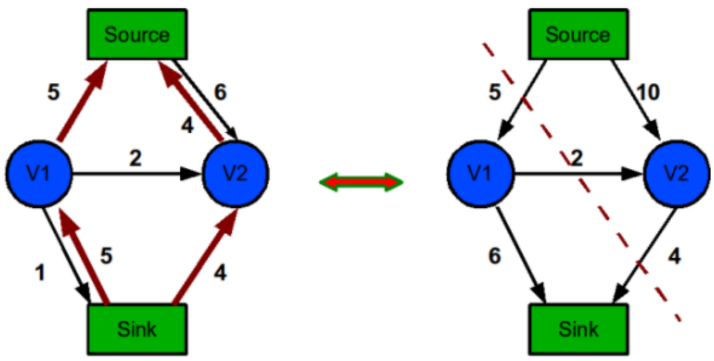
- *f is a max flow in G*
- *The residual network Gf contains no augmenting path*
- *There exist a cut C(S, T) with capacity f*



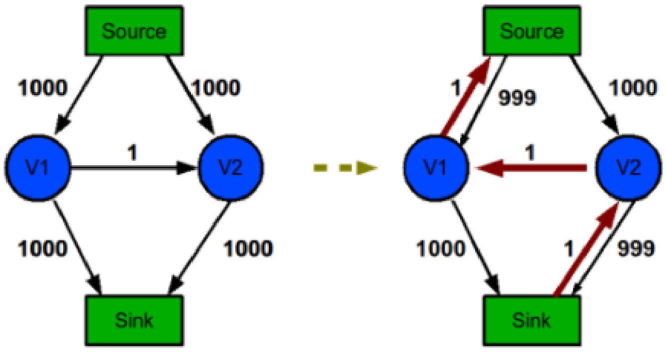
In finding the augmented path p for the above graph the flow is identified as 0



Here the flow is 5



Here the flow is 5+4 =9
Thus $max - flow = |f*|$ *then at worst case the complexity of the algorithm based on Ford-Flukerson:* $O(|f*||E|)$
on Ford-Flukerson Method



If the traverse is done this way, then the edges have to be traversed 2000 times

Edmonds-Karp Algorithm finds the augmenting path with a breadth first search and has a complexity of $O(|V||E|^2)$