# Week 09

## Graph Theory & Its Application

## Previous Class Discussion

### 9.1 Assignment Problem as Optimization (Hungarian Algorithm)

C = [ cij ] nxn $x_{ij}$ = 1 person i is assigned task j.

$$\Sigma_i x_{ij} = 1 \ \& \ \Sigma_j x_{ij} = 1$$

$$\text{Minimise} \quad \Sigma_i \Sigma_j C_{ij} x_{ij}$$

## Previous Class Discussion End

### 9.2 Calculation of Dual of optimization Problem

We will try to do dual of optimization Problem.

As we the Matrix C, then we find the Row minimum and subtract and then find the column minimum and subtract

Let $u_1, u_2 \ldots u_n$ be Row minimum and $v_1, v_2 \ldots$ & $v_n$ be the Column Minimum.

So, the C' = []

$C'_{ij}$ can be written as $C'_{ij} = C_{ij} - (u_i + v_j)$

$C'_{ij} \geq 0 \Longrightarrow C_{ij} - (u_i + v_j) \geq 0$ that means $C_{ij} \geq (u_i + v_j)$ ---------- Eq (1)

If we write the previous optimization problem in dual form then,

Previous optimization problem:

$$\text{Minimise} \quad \Sigma \Sigma C_{ij} x_{ij}$$

$$\text{Such that} \quad \Sigma_i X_{ij} = 1 \text{ and}$$

$$\Sigma_j X_{ij} = 1 \text{ and } X_{ij} \geq 0$$

Now we will take some additional variable that is called $u_i$ , $v_j$ is Dual Variable

$$\Sigma_i x_{ij} = 1 \qquad u_i \ \forall \ i = 1 \ldots n$$

$$\Sigma_j x_{ij} = 1 \qquad v_j \ \forall \ j = 1 \ldots n$$



Lecturer: Anand Mishra, Scribe: Devbrat Kumar-M20AIE232

Now, $(u_1 + v_1)x_{11} + (u_2 + v_2)x_{22} + (u_1 + v_2)x_{12} .... (u_n + v_n)x_{nn} = u_1 + u_2 + .... + u_n + v_1 + v_2 + .... + v_n$

$$\Rightarrow \quad \Sigma_i \Sigma_j (u_i + v_j)x_{ij} = \Sigma_i u_i + \Sigma_j v_j$$

We can write this in simpler way,

$$\Sigma u_i + \Sigma v_j = \Sigma \Sigma (u_i + v_j)x_{ij}$$

That means,

$$\Sigma u_i + \Sigma v_j \leq \Sigma \Sigma C_{ij} x_{ij}$$

Now we have to minimise $\Sigma \Sigma C_{ij} x_{ij}$

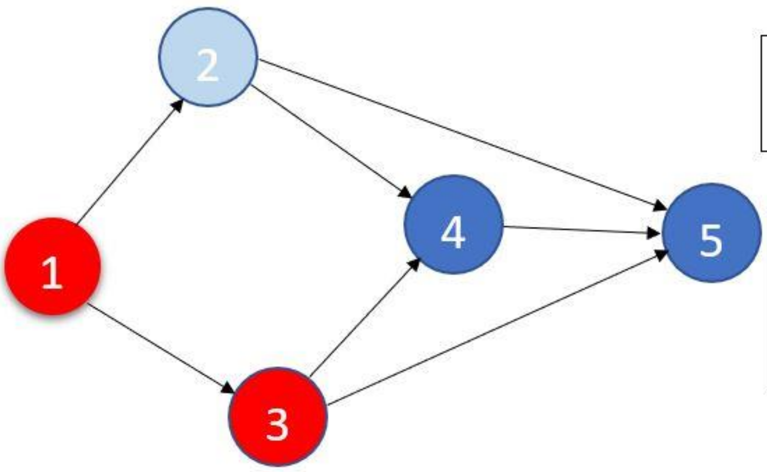
In Other Words, we can say that, maximise $\Sigma u_i + \Sigma v_j$ instead of minimise other

Now, Dual Optimization Problem is,

Maximise $\Sigma_i u_i + \Sigma_j v_j$

Subject to $C_{ij} \geq u_i + v_j$

## 9.3 Minimum Cost Flow Network




$b_1 = 6$, $b_2 = 0$,
$b_3 = 4$, $b_4 = -5$, $b_5 = -5$

$C_{12} = 8$, $C_{13} = 6$, $C_{24} = 5$
$C_{34} = 6$, $C_{25} = 7$, $C_{35} = 3$
$C_{45} = 4$

Problem: we have Red Nodes (Supply Nodes) where production happens, Blue Nodes (Demand Nodes) where consumption happens, Cyan Node (Intermediate Nodes).

$b_1 = 4$ means it produces 4 units, $b_4 = -5$ means it consumes 5 units

Note: total unit of Demand will be equal to total unit of consumption.

Cost on Edges denotes transport cost per unit from one node to other

$C_{12}$=8 means, if you transport 1 unit of item from Node 1 to Node 2, cost is 8.

Q. Find out the minimum cost flow for transporting all the materials from supply nodes to demand nodes.

Now Optimization problem correspondence to above problem,

$$\text{Minimise} \quad \Sigma_i \Sigma_j C_{ij} x_{ij}$$

$$\begin{aligned} \text{Subject to} \quad & x_{12} + x_{13} = b_1 \\ & -x_{12} + x_{24} + x_{25} = b_2 \\ & -x_{13} + x_{34} + x_{35} = b_3 \\ & -x_{24} - x_{34} + x_{45} = b_4 \\ & -x_{25} - x_{45} + x_{35} = b_5 \end{aligned}$$

Now convert the above problem in dual problem,

We will introduce the new variable called dual variable, i.e., $w_1$, $w_2$, $w_3$, $w_4$ & $w_5$ that will multiply with above equation and then sum it

We have,

$$(w_1 - w_2)x_{12} + (w_1 - w_3)x_{13} + (w_2 - w_4)x_{24} + \ldots\ldots.. = \Sigma W_i b_i$$

Now It can be written as :

$$\Sigma_i \Sigma_j (W_i - W_j)x_{ij} = \Sigma W_i b_i$$

If we apply following constraints,

$$W_i - W_j \leq C_{ij} \text{ then}$$

$$\Sigma_i W_i b_i = \Sigma_i \Sigma_j (W_i - W_j) x_{ij} \leq \Sigma C_{ij} x_{ij}$$

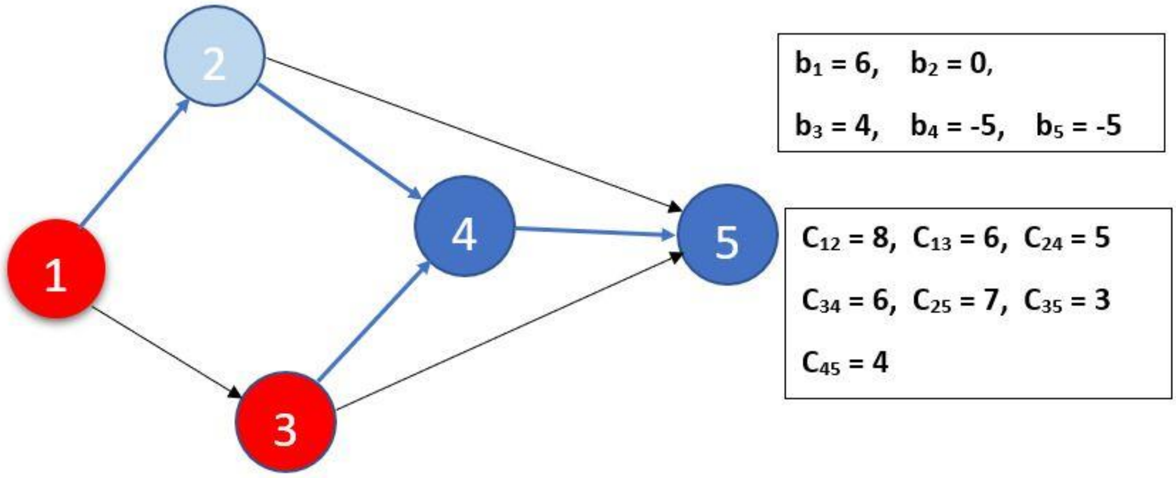
Original Problem was to minimise $\Sigma C_{ij} x_{ij}$ so in other words we will maximise $\Sigma_i W_i b_i$

So equivalent problem will be

$$\text{Max} \quad \Sigma W_i b_i$$

$$\text{s.t.} \quad W_i - W_j \le C_{ij} \qquad \text{or} \qquad W_i - W_j - C_{ij} \le 0$$

## 9.4 Solution of Above Optimization Problem

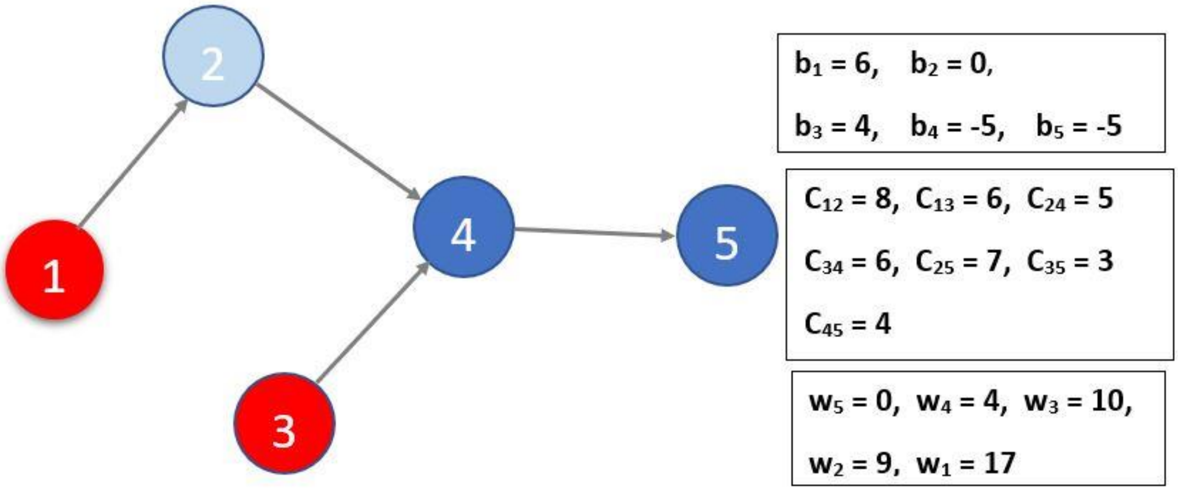



Note: Blue Line is Path for Minimum Cost

Cost = 6*8 + 6*5 + 4*5 + 6*4 = 48 + 30 + 20 + 24 = 122

Now We Will find the above solution is good solution or can have better solution.

To find another solution we will introduce another variable W5 = 0



Since we know, $W_i - W_j = C_{ij}$ or $W_i = Wj + C_{ij}$

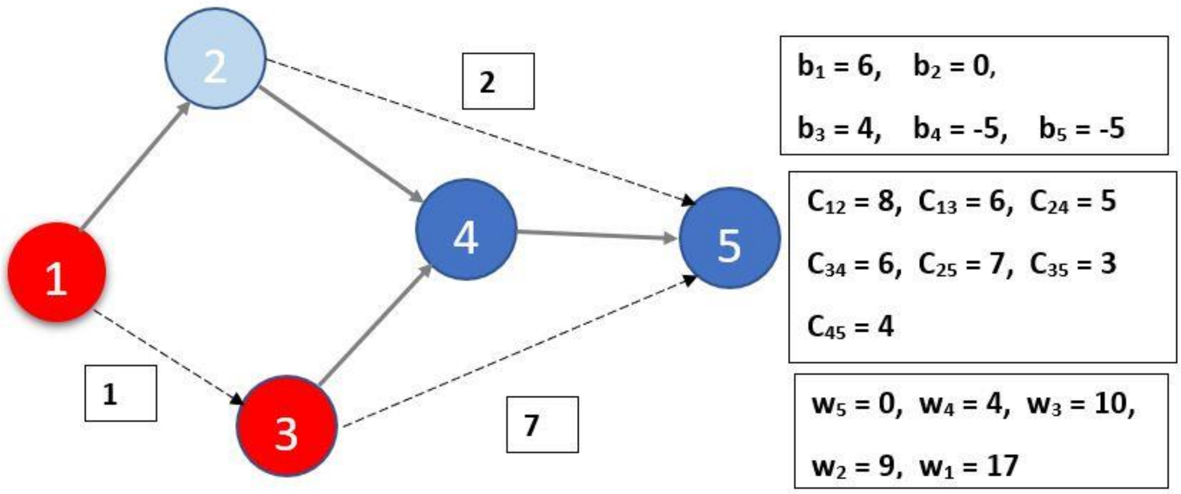
Since $W_5 = 0$, so $W_4 = 4$, $W_2 = 9$, $W_3 = 10$ & $W_1 = 17$

Now We will verify for Non basic Edges (Edges Not Part of Solution), if $W_i - W_j - C_{ij}$ is negative or not

For Edge 2-5, $W_{25} = W_2 - W_5 - C_{25} = 9 - 0 - 7 = 2$

Similarly for Edge 3-5, $W_{35} = W_3 - W_5 - C_{35} = 10 - 0 - 3 = 7$

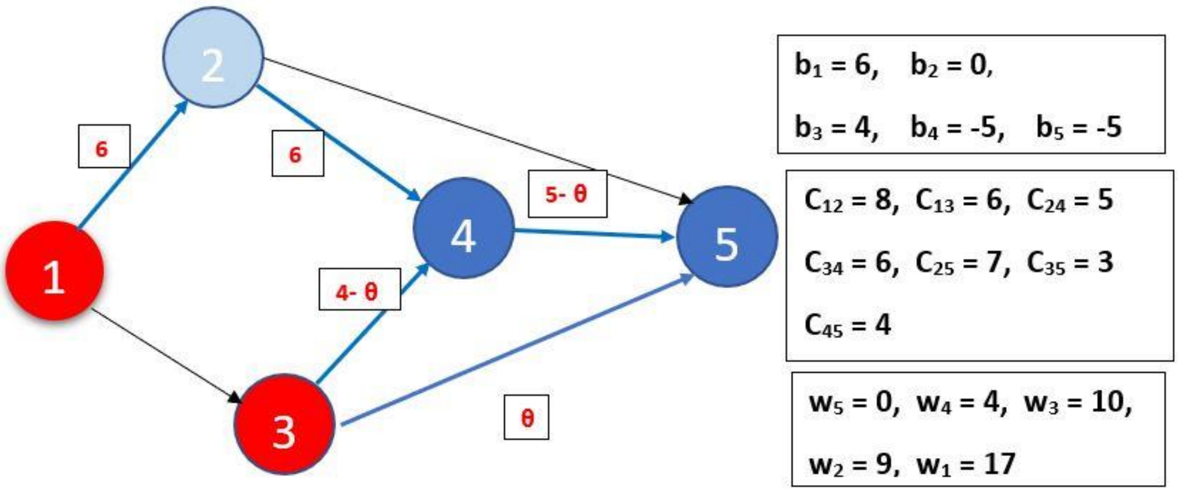
Similarly for Edge 1-3, $W_{13} = W_1 - W_3 - C_{13} = 17 - 10 - 6 = 1$




Since It does not follow the constraints as all the values is positive that means the solution is not optimal solution so we have to modify the solution to get optimal solution

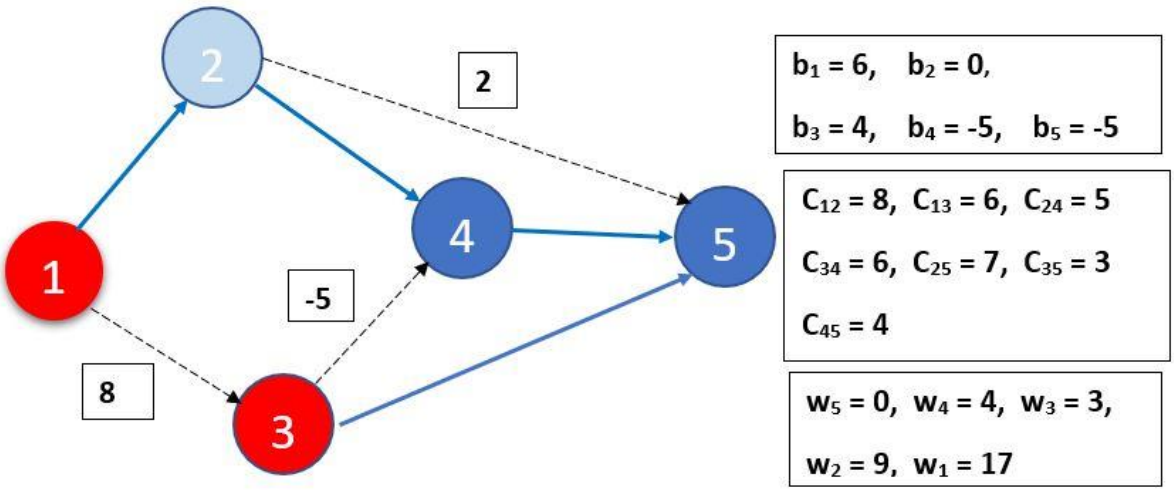
To find the optimal solution we will include the edges that is most violating i.e Edge 3-5

Hence another solution will be




Since Node 3 has two transport edge, so from one edge it will transport θ and rest 4-θ

Let’s assume maximum value of θ




Note: Since Cost for Edge 3-4 is 0 so it will not part of solution.

Cost = 6*8 + 6*5 + 4*3 + 1*4 = 48 + 30 + 12 + 4 = 94

Still, we don’t know that the solution is optimal solution.

To Know, whether it is optimal solution or not, we will follow the same for non-basic edges

Since $W_5 = 0$, so $W_4 = 4$, $W_2 = 9$, $W_3 = 3$ & $W_1 = 17$

For Edge 2-5, $W_{25} = W_2 - W_5 - C_{25} = 9 - 0 - 7 = 2$

Similarly for Edge 3-4, $W_{34} = W_3 - W_4 - C_{34} = 3 - 4 - 6 = -7$

Similarly for Edge 1-3, $W_{13} = W_1 - W_3 - C_{13} = 17 - 3 - 6 = 8$

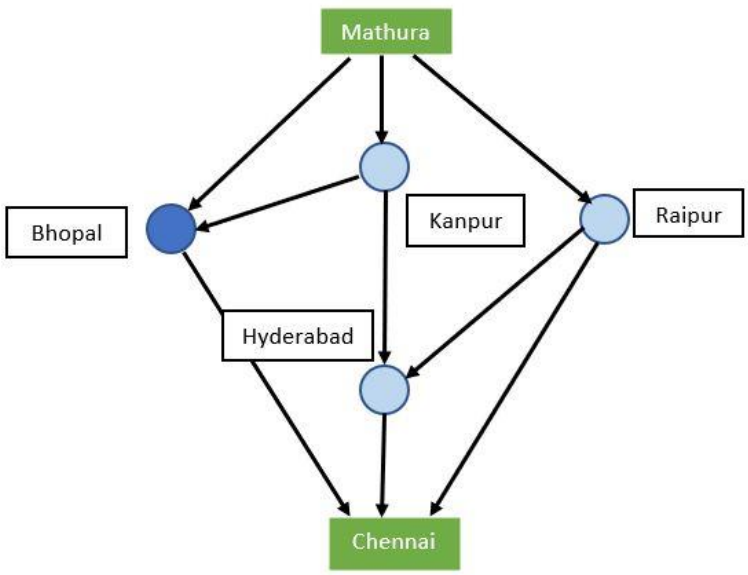
Since it does not contain all negative values means this is not optimal solution

Hence, we will include Edge 1-3 with $\theta$ transport.

After repeating We will get Optimal Solution as 81.

## 9.5 Maximum Flow Problem

Motivation: Imagine a oil refinery at source (Mathura) producing oil and it has warehouse in Chennai. There are multiple path from source (Mathura) to destination (Chennai) with each path having some capacity of fluid flow. Such graph is known as flow network.




**Cost:**

**Mathura -> Kanpur: 900**

**Mathura -> Bhopal: 1000**

**Mathura -> Raipur: 600**

**Bhopal -> Chennai: 600**

**Kanpur -> Bhopal:**

**Kanpur -> Hyderabad: 200**

**Raipur -> Hyderabad: 400**

**Raipur -> Chennai: 1000**

**Hyderabad -> Chennai: 300**

Note: Edges represent pipes and number represent the capacity that can flow.

## 9.6 Difference between Maximum flow and Minimum Cost flow

1. Instead of capacity they have per unit cost
2. There were demand and supply nodes and these nodes do not have infinite capacity.
3. There will be one source and destination node but in min cost floe there were multiple demand and supply nodes.

## 9.7 Problem can be modelled as flow network

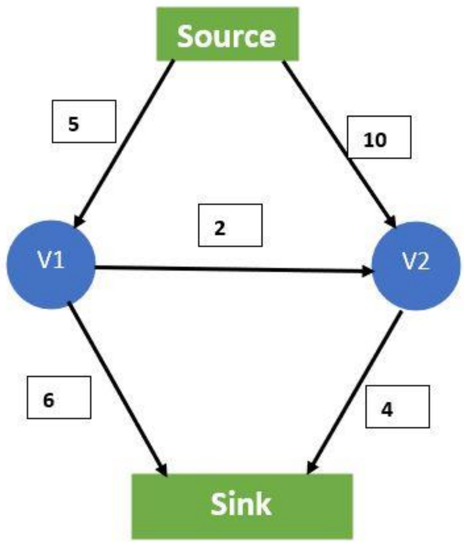
1. Liquids flow through pipes
2. Current through electrical networks
3. Information through communication network
4. Vehicles through roads

In Flow Network:

1. Each vertex other than source and destination/sink is a conduit junction. They do not store/collect any materials.
2. Each Edge can be though of a conduit for the material with a predefined capacity.

## 9.8 Terminology:

**Flow Network**: a flow network G = (V, E) is a directed graph in which each edge (u, v) ∈ E has no negative capacity c(u, v) ≥ 0



A flow network is always a connected graph thus in any flow network, $|E| \geq |V| - 1$.

Sometime it is also called s-t graph s stands for source and t stands for sink.

**Flow**: Let G = (V, E) be a flow network with a capacity function c, lets s be the source of the network and t be the sink. then a flow in G is defined as a real valued function $f: V \times V \rightarrow R$ that satisfies the following three properties

Capacity Constraint: $f(u, v) \leq c(u, v) \quad \forall u, v \in V$

Skew Symmetry: $f(u, v) = -f(v, u) \quad \forall u, v \in V$

Flow Conservation: $\Sigma\, u \in V\, f(u, v) = 0 \;\; \forall u \in V - \{s, t\}$

Or equivalently $\Sigma\, v \in V\, f(v, u) = 0 \;\; \forall v \in V - \{s, t\}$

Total Net Flow: Total positive flow entering a vertex v is defined as,

$\Sigma\, u \in V, f(u, v) > 0\; f(u, v)$

Similarly, we can define the total positive flow leaving vertex v as

$\Sigma\, v \in V, f(v, u) > 0\; f(v, u)$

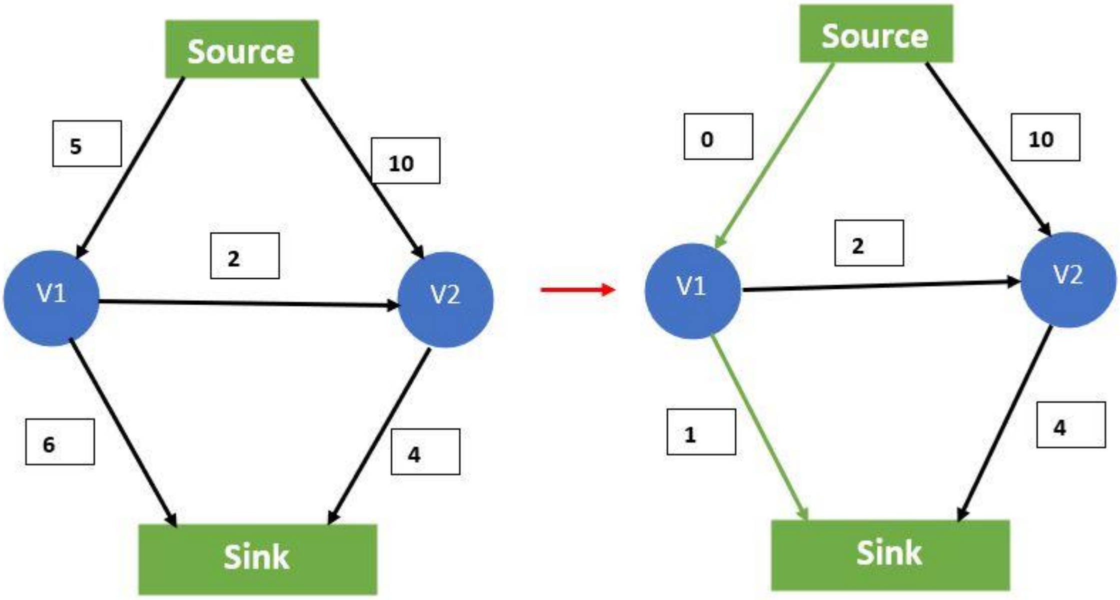
Hence total net flow of vertex v s defined as total positive flow leaving vertex v minus total positive flow entering that vertex.

**Residual Network and Residual Capacity**:

Residual Network: Given a flow network G = (V, E) and flow f the residual network of G induced by flow f is

$G_f = (V, E_f)$ where $E_f = \{(u, v) \in V \times V : C_f(u, v) > 0\}$

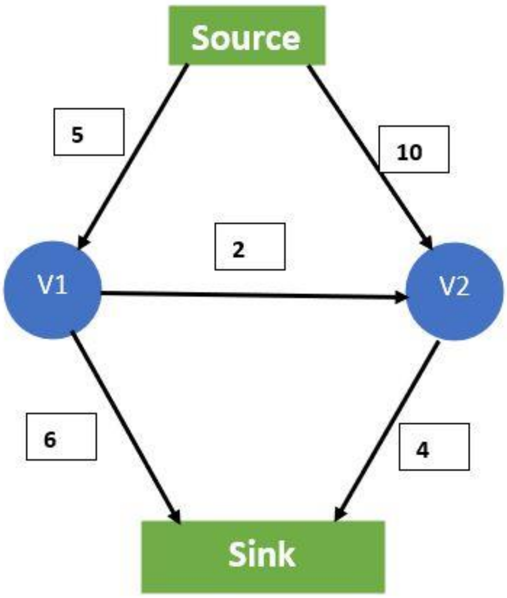
Residual Capacity: The amount of flow we can push from u to v before exceeding the capacity c(u, v) is the residua capacity of c(u, v)




Note: Right hand side graph is Residual Network/Graph

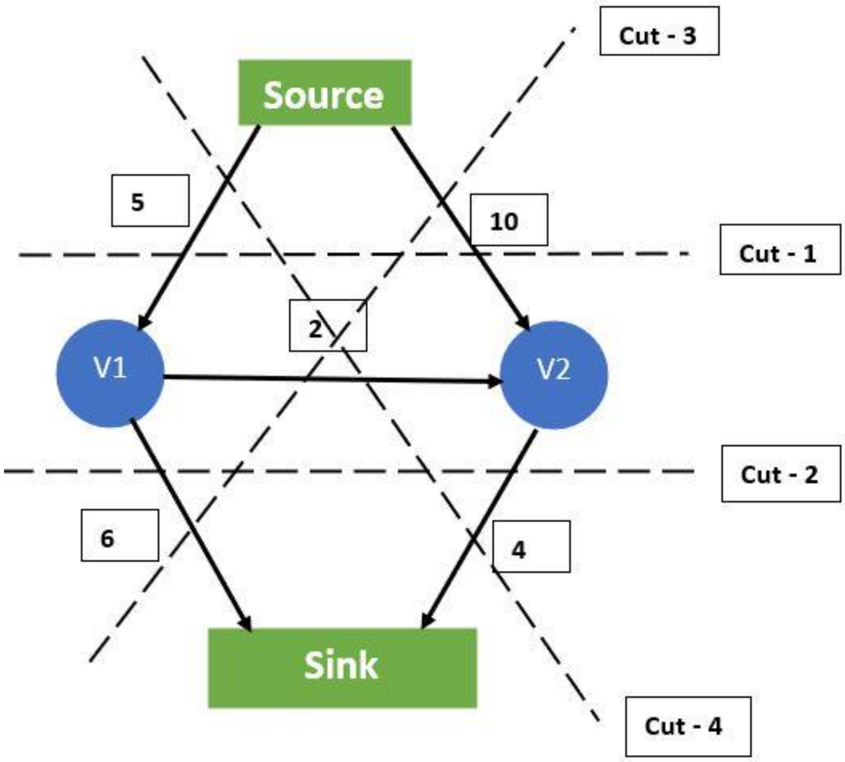
**Augmenting Path**: Given a flow network G = (V, E) and a flow f an augmenting path p is a simple path from s to t in the residual network.

Residual capacity of path is the minimum residual capacity along the path.



| | | |
|---|---|---|
| Possible Path: | Source → V1 → Sink : | Residual capacity of Path = 5 |
| | Source → V1 → V2 → Sink: | Residual capacity of Path = 2 |
| | Source → V2 → Sink: | Residual capacity of Path = 4 |

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




Cut 1- S: {source}, T: {V1, V2, sink} & Capacity of Cut 1: 15

Cut 2- S: {V1, V2, source}, T: {sink} & Capacity of Cut 2: 10

Cut 3- S: {V1, source}, T: {V2, sink} & Capacity of Cut 3: 18

Cut 4- S: {V2, source}, T: {V1,sink } & Capacity of Cut 4: 9 (Note: Edges from T Side to S Side will not be included).

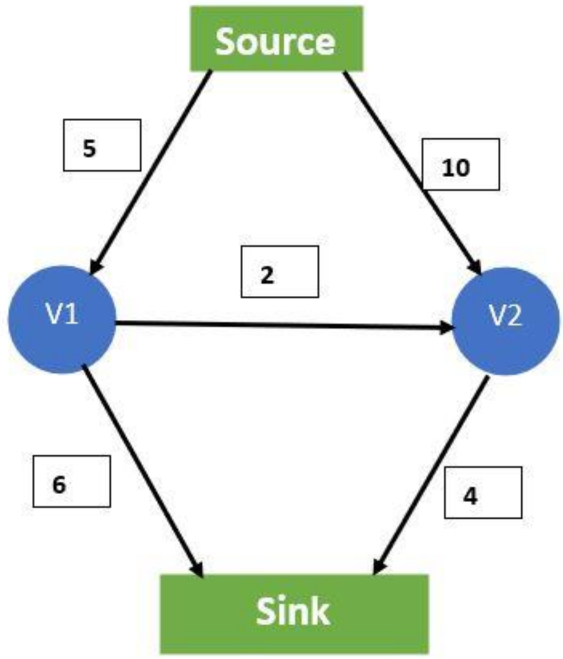
Min Cut : 9

## 9.9 Max Flow – Min Cut Theorem

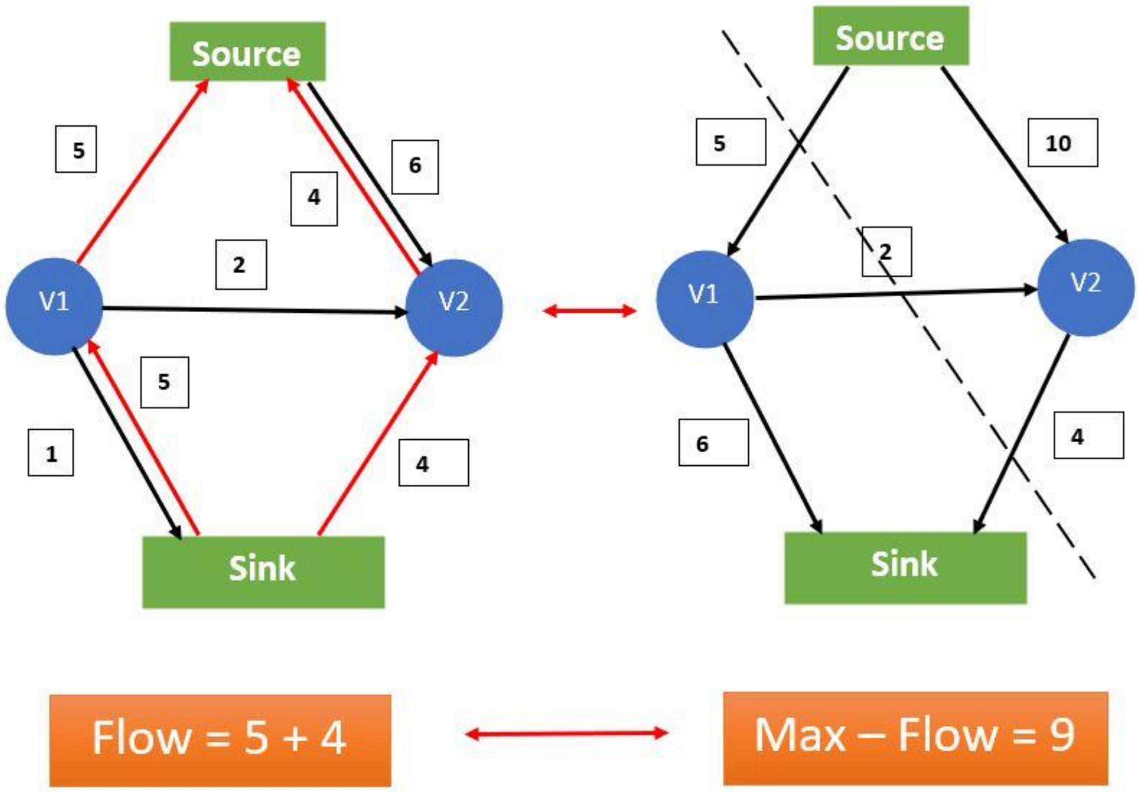
If f is a flow in network G = (V, E) with sources s and sink t, then the following conditions are equivalent:

f is a max flow in G then the residual network $G_f$ contains no augmenting path there exist a cut C(S, T) with capacity f an that cut is minimum cut.

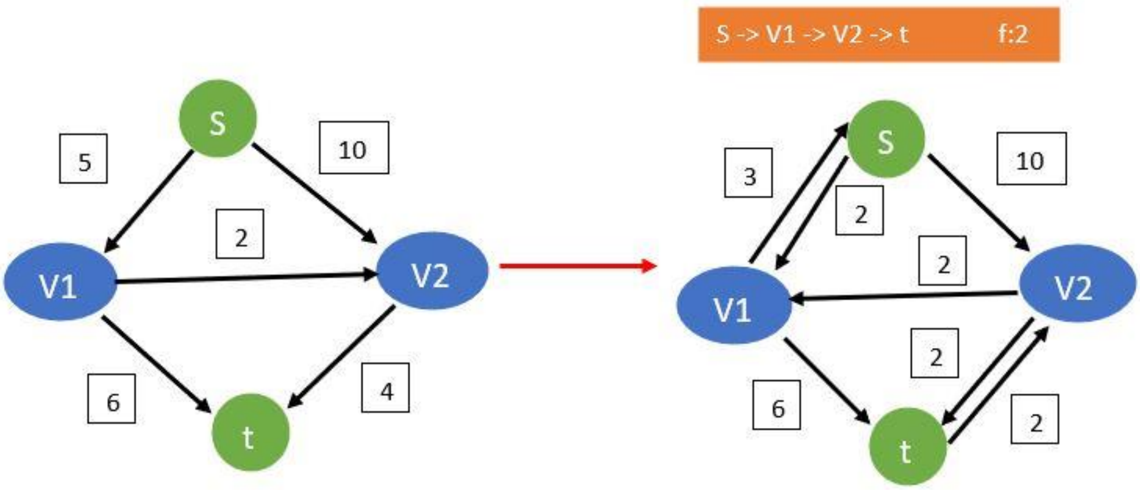
## 9.10 Ford-Flurkerson Method

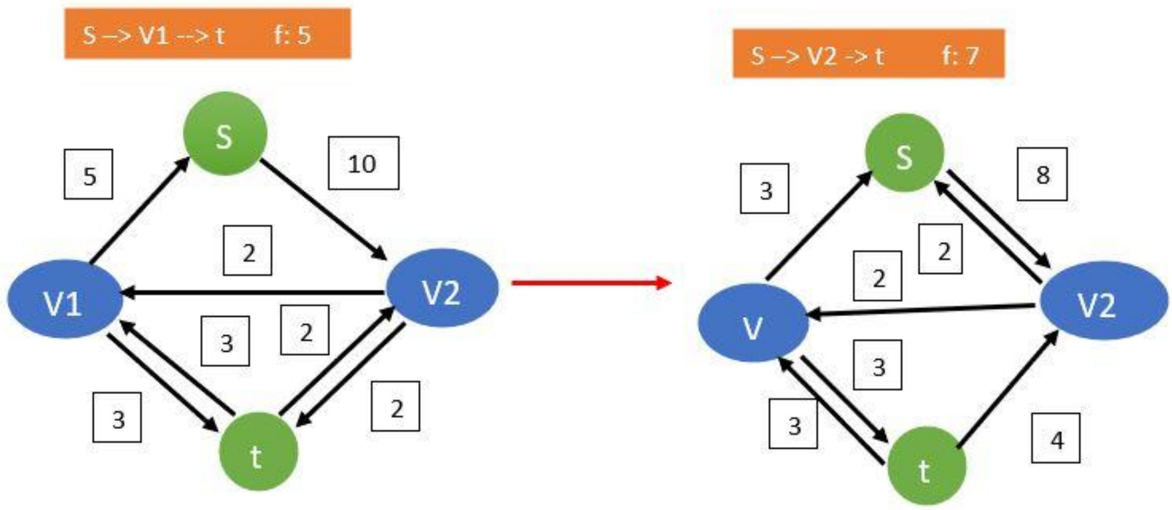


Find an augmenting path p and augment flow f against p, if there is no augmenting path then stop.




Another Possible Path






**Analysis of Algorithm**:

While finding augmenting path one has to traverse O(|E|) each time. Thus if max – flow = |f*| then at worst case the complexity of the algorithm based on ford-flukerson: O(|f*| |E|)

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