Chapter 9

Week : 09
Minimum Cost Flow Network, Graph Cut, Ford-Flukerson Method*

Let, we have a cost matrix $C = [c_{ij}]_{n\times m}$ and $x_{ij} = 1$ means 1 person i is assigned task j and also $\sum_i x_{ij} = 1$ and $\sum_j x_{ij} = 1$ and ultimately we want to minimize $\sum_i \sum_j c_{ij}x_{ij}$. Now, this is our problem and now we will try to obtain a optimal solution for this.

Previously we found row minimum and substract and column minimum and substract. Suppose we have the matrix, $C' =$

$$\begin{array}{cc} \begin{array}{ccccc} u_1 & u_2 & u_3 & \dots & u_n \end{array} & \\ \left[\begin{array}{ccccc} & & & & \\ & & & & \\ & & c_{ij} & & \\ & & & & \\ & & & & \end{array}\right] & \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ \dots \\ v_n \end{array} \end{array}$$

After subtraction of row minimum and column minimum we get,

$$C'_{ij} \;=\; C_{ij} - (u_i + v_j)$$

And u_i and V_j are chosen such that,

$$C'_{ij} \geq 0$$

i.e.,

$$C_{ij} \;\geq\; u_i + v_j$$

.

Now, our previous optimization problem is changed to a dual problem. Now, we write our original problem,

*Lecturer: Anand Mishra. Scribe: Debanshu Biswas(M20MA053).

$$\min \sum_i \sum_j c_{ij} x_{ij}$$
$$\text{s.t. } \sum_i x_{ij} = 1$$
$$\sum_i x_{ij} = 1$$
$$x_{ij} \geq 0$$

We will take some additional variables u_i $\forall i = 1, 2, 3, \ldots, n$ and v_j $\forall j = 1, 2, 3, \ldots, n$ which are called dual variables and rewrite the same problem and make the problem a maximization problem.So, in total we will introduce $2n$ variables. And they are related to x_{ij} as follows,

$$(u_1 + v_1)x_{11} + (u_2 + v_2)x_{22} + (u_3 + v_3)x_{33} + \cdots + (u_n + v_n)x_{nn} =$$
$$u_1 + u_2 + u_3 + \cdots + u_n + v_1 + v_2 + v_3 + \cdots + v_n$$
$$\Rightarrow \sum_i \sum_j (u_i + v_j)x_{ij} = \sum_i u_i + \sum_j v_j$$
$$\text{Also, } c_{ij} \geq u_i + v_j$$
$$\Rightarrow \sum_i \sum_j (u_i + v_j)x_{ij} = \sum_i u_i + \sum_j v_j \geq \sum_i \sum_j c_{ij} x_{ij}$$
$$\Rightarrow \sum_i u_i + \sum_j v_j \leq \sum_i \sum_j c_{ij} x_{ij}$$

Now, we can write the dual problem as,

$$\max \sum_i u_i + \sum_j v_j$$
$$\text{s.t. } u_i + v_j \leq c_{ij}$$

9.1 Minimum cost flow network problem

Now, we will look into minimum cost flow network problem. The problem will be described using the following graph. As, shown in the figure in this problem we have to transport some goods from two supply units, i.e., red nodes to two demand nodes, i.e., blue nodes according to their demands and supply capacity denoted by b_i. There is also a intermediate node. And, the cost is given by c_{ij}. And the problem is,

Problem : Find out minimum cost flow for transporting all the material from supply nodes to demand node.

Now, we write the above minimum cost flow problem mathematically then the optimization formulation is,

$$\min \sum_i \sum_j c_{ij} x_{ij}$$
$$\text{Subject to, } x_{12} + x_{13} = b - 1$$
$${}_{12} + x_{24} + x_{25} = b_1$$
$$-x_{13} + x_{34} + x_{35} = b_2$$
$$-x_{24} - x_{34} + x_{45} = b_4$$
$$-x_{25} - x_{45} + x_{35} = b_5$$

Now, we will convert this problem in dual in variables u_i and v_j. This way number of variables will decrease. We, will multiply above five constrains by w_1, w_2, w_3, w_4 and w_5 respectively and sum all. Then we will get,

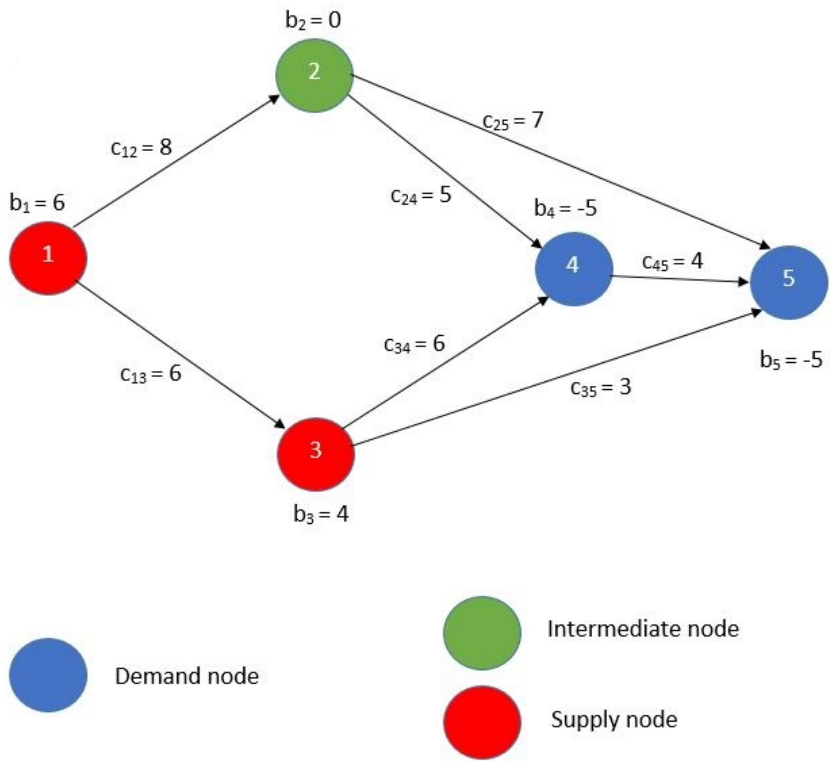

Figure 9.1: Minimum cost flow network problem

$$(w_1 - w_2)x_{12} + (w_1 - w_3)x_{13} + (w_2 - w_4)x_{24} + \cdots = \sum_i w_i b_i$$
$$\Rightarrow \sum_i \sum_j (w_i - w_j)x_{ij} = \sum_i w_i b_i$$

Now, lest us apply the following constraints,

$$w_i - w_j \leq c_{ij}$$
$$\sum_i w_i b_i = \sum_i \sum_j (w_i - w_j)x_{ij} \leq \sum c_{ij}x_{ij}$$

So, the dual problem becomes,

$$\max \sum_i w_i b_i$$
$$\text{subject to } w_i - w_j \leq c_{ij}$$

Now, we consider a initial basic feasible solution for this problem. We will show this by figure 9.2.

Here, we will follow the blue path to transport items. And in those blue edges we wrote the capacity of items transported in red bold colour.

This way, we get total cost,

Cost $= 6 \times 8 + 6 \times 5 + 4 \times 5 + 6 \times 4$

$=48 + 30 + 20 + 24$

$= 122$

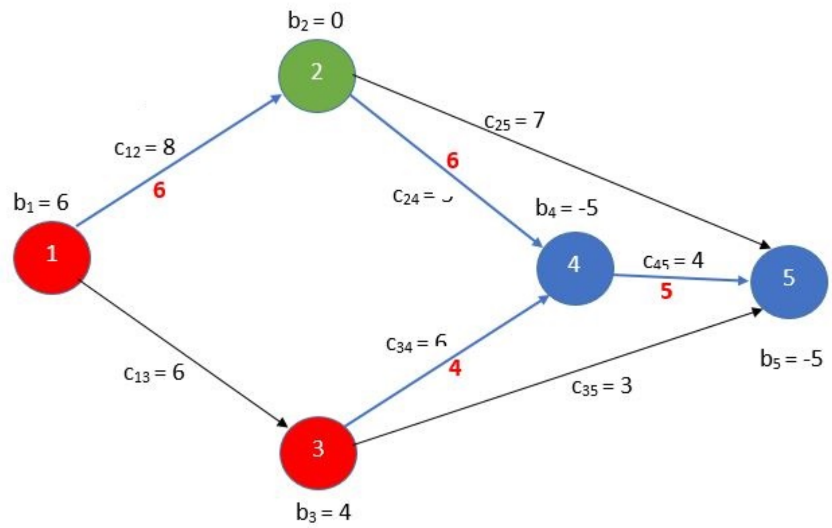

Figure 9.2: Minimum cost flow network problem initial basic feasible solution

Now, we will improve this solution to get optimal solution. And, for this we will introduce dual variables as shown in figure 9.3.

And for introducing dual variables we will ensure that $w_i - w_j = c_{ij}$. And, for non basic variables we will ensure that, $w_i - w_j - c_{ij}$ is negative. Now, we will check for other edges. And if this is not satisfied then we will improve. All the values of $w_i - w_j - c_{ij}$ for non basic variables are given in figure 9.4. We, observe that all are positive. So, we need to improve.

To modify we will include that edge that violating the most. So, we include edge 35. We assume that we will transport in this edge. So, This also changes that in edge we will transport $4-$ and in edge 45 we will transport $5-$.

In this case we take maximum thita, i.e., $= 4$. This is given in figure 9.5.this case cost is,
$=6 \times 8 + 6 \times 5 + 4 \times 3 + 1 \times 4$
$=48 + 30 + 12 + 4$
$=94$

So, here the cost reduced little bit.

Now we will check for optimality. This is given in figure 9.6. Here also we are getting positive 8 at edge 13. Now we have to include the edge 13. Then we take $= 6$. And, calculating tn the same way we will get optimal cost =81. And the last case will be optimal solution.

This method is called network simplex method.

9.2 Graph Cut, Maximum flow problem

Imagine a oil refinery at Mathura producing oil and it has a warehouse in Chennai. There are multiple path from the source (Mathura) to destination(Chennai) with each path having

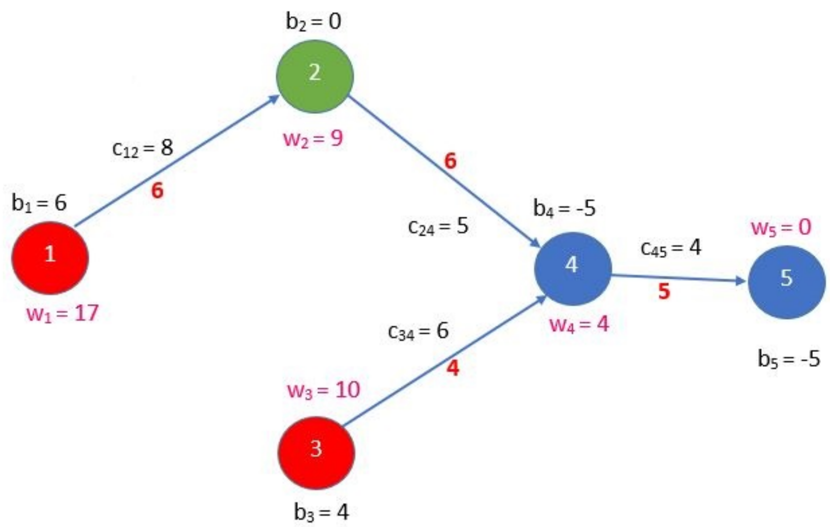

Figure 9.3: Defining dual variables

some capacity of fluid flow. Such graph are known as flow network. And, this is shown in the figure 9.7.

The following problems can be modelled as flow network.

- Liquids following through pipes
- Current through electrical networks
- Information through communication networks
- Vehicles through roads

The flow network can not hold items. It always flows. In a flow network

- Each vertex other than source and sink is a conduit junction. They do not store/collect any material.(In context of electrical networks this is the well known rule: Kirchoff's Law)
- Each edge can be though of a conduit for the material with a predefined capacity. For example: 100 gallon liquid per hour through a pipe or 10 amperes current through a wire.

now, we will investigate what is the maximum flow possible in a given flow network problem. In this regard we will first see definitions of flow network, flow and max-flow problem.

Definition 9.1 (Flow network)**.** A flow network $G = (V, E)$ is a directed graph in which each edge $(u, v) \in E$ has a non-negative capacity $c(u, v) \geq 0$.

In a flow network we distinguish two vertices source(s) and sink(t).

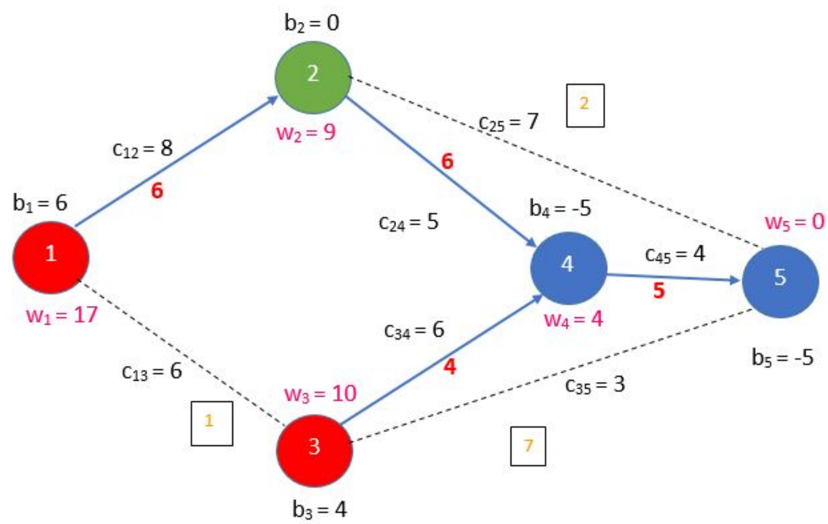

Figure 9.4: Checking constraints for non basic variables

The graphical interpretation in given in figure 9.8.

We note that a flow network is always a connected graph thus in any flow network,

$$|E| \geq |V| - 1$$

Definition 9.2 (Flow)**.** Let $G = (V, E)$ be a flow network with a capacity function c, let s be the source of the network and t be the sink. Then a flow in G is defined as a real valued function $f : V \times V \to \mathbb{R}$ that satisfies following three properties:

- Capacity constraint: $f(u, v)(u, v); \quad \forall u, v \in V$
- Skew Symmetry: $f(u, v) = f(v, u); \quad \forall u, v \in V$
- Flow Conservation:

  $\sum_{u \in V} f(u, v) = 0; \quad \forall u \in V\{s, t\}$

  or equivalently, $\sum_{v \in V} f(v, u) = 0; \quad \forall u \in V\{s, t\}$

Definition 9.3 (Total net flow)**.**
- Total positive flow entering a vertex v is defined by: $\sum_{u \in V,\ f(u,v)>0} f(u, v)$
- Similarly, We can define total positive flow leaving a vertex v as: $\sum_{v \in V,\ f(v,u)>0} f(v, u)$
- Total net flow of vertex v is defined as total positive flow leaving vertex v minus total positive flow entering that vertex.

Problem : Given a flow network G with source s and sink t we wish to find a flow of maximum value.

Now we will see definitions of residual network and residual capacity.

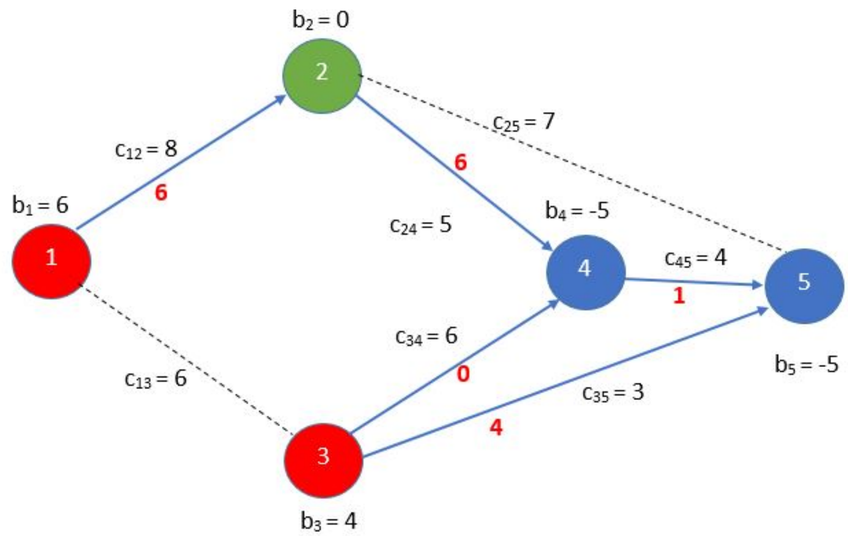

Figure 9.5

Definition 9.4 (Residual Network)**.** Given a flow network $G = (V, E)$ and a flow f the residual network of G induced by flow f is $G_f = (V, E_f)$ where

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

Definition 9.5 (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 $c(u, v)$.

These two definitions can be visualized in figure 9.9.

Definition 9.6 (Augmenting path)**.** Given a flow network G=(V,E) and a flow f an augmenting path p is a simple from s to t in the residual network.

Residual capacity of path: is the minimum residual capacity along the path.
This is portrayed in figure 9.10. Residual of the path here is 2.

Definition 9.7 (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. This is shown in figure 9.11. And, capacity of this cut is 10.

Capacity of cut in figure 9.12 is 9.

Theorem 9.8 (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.*

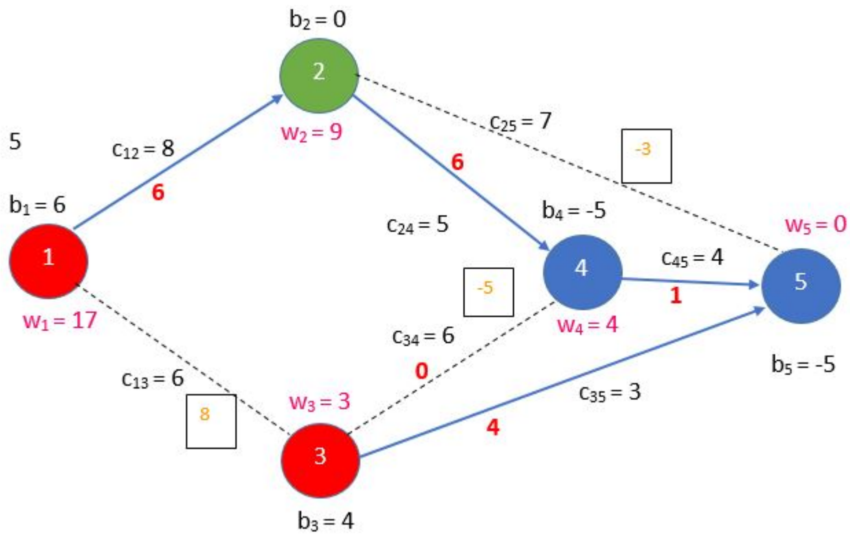

Figure 9.6

9.3 Ford-Flukerson Method

Now, we will discuss **Ford-Flukerson Method**. The steps are as follows.

- Find an augmenting path p and augment flow f against p.
- proceed until no augmented path exists.
- Now, find a cut such that we can max-flow.

This method is described for an example in graph 9.13.

Now, we analyze Ford-Flukerson Method. While finding augmenting path one has to traverse $O(|E|)$ each time. Thus if $maxflow = |f|$ then at worst case the complexity of the algorithm based on Ford-Flukerson: $O(|f||E|)$.

The worst complexity case can be seen in figure 9.14.

We will now discuss about Efficient Ford-Flukerson Method: Edmonds-Karp Algorithm.

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

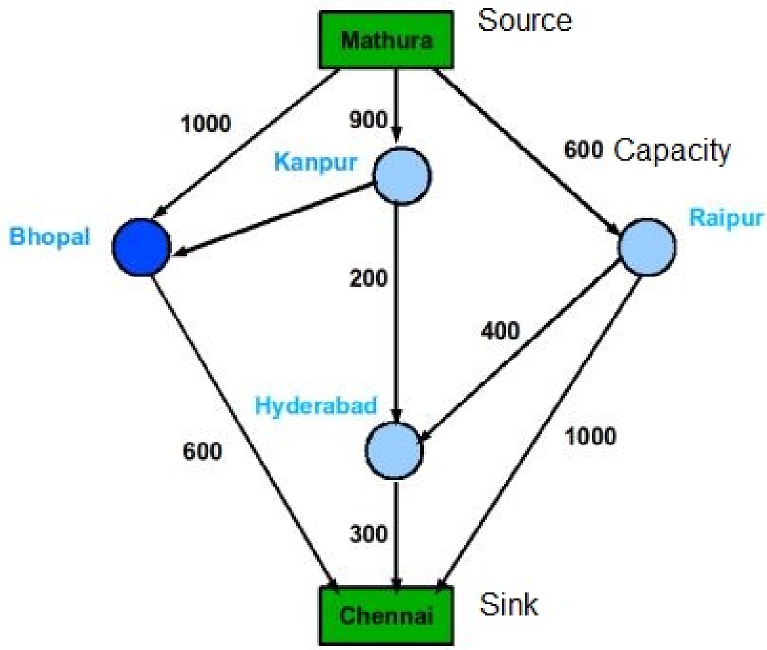
Mathura
Source
1000
900
600 Capacity
Kanpur
Raipur
Bhopal
200
400
Hyderabad
1000
600
300
Chennai
Sink

Figure 9.7

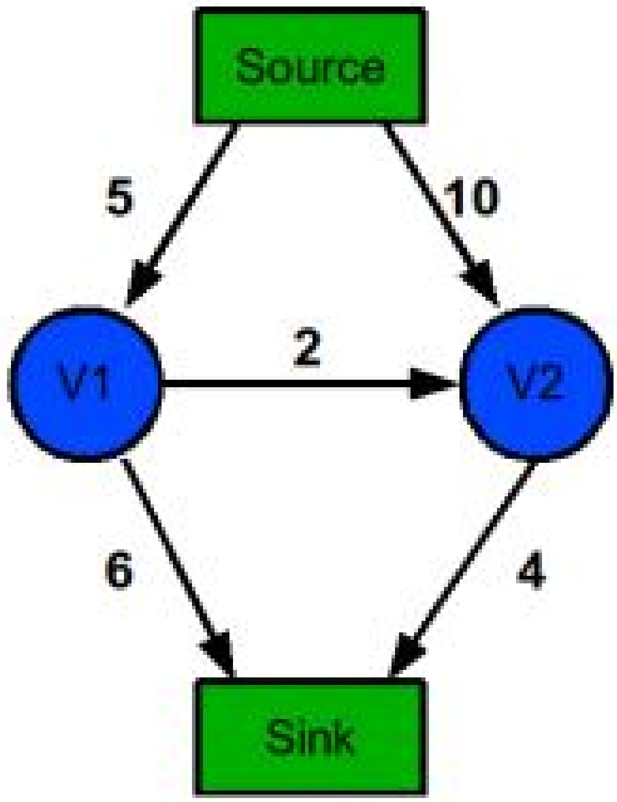

Figure 9.8: Flow network

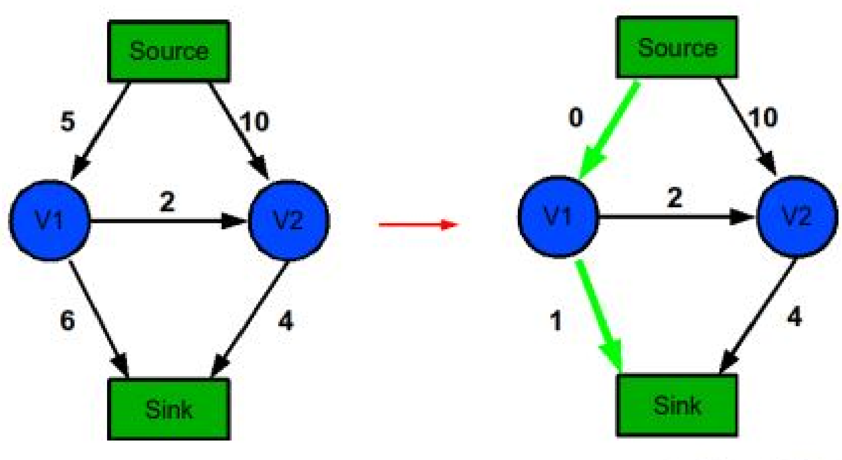

Figure 9.9: Residual network and capacity

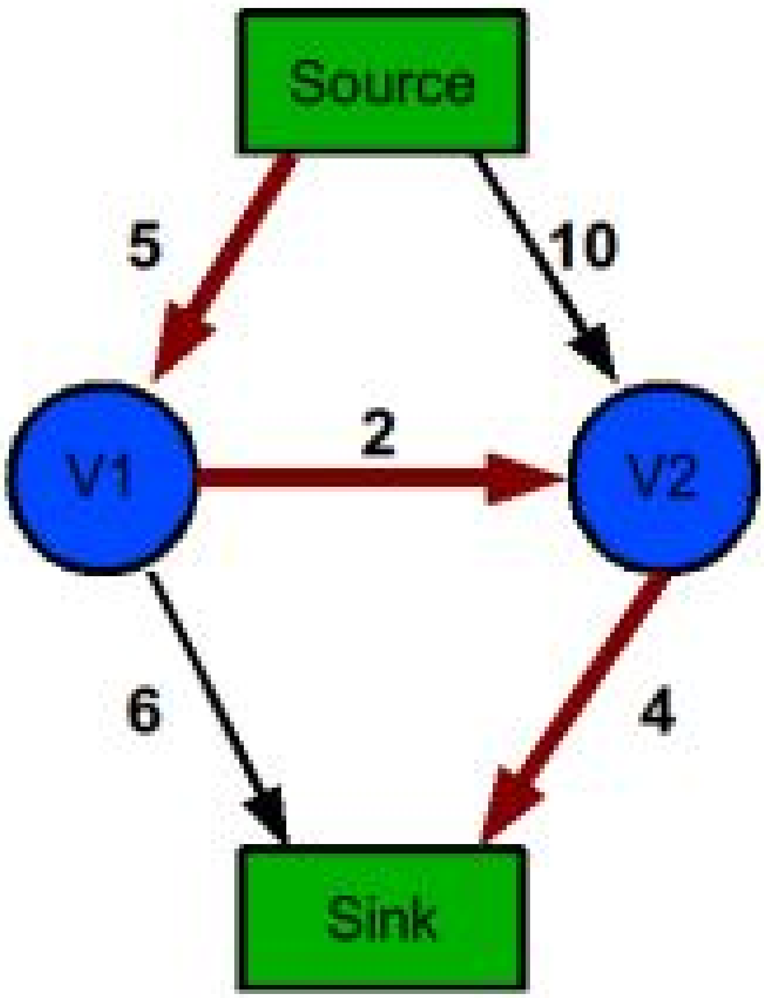

Figure 9.10

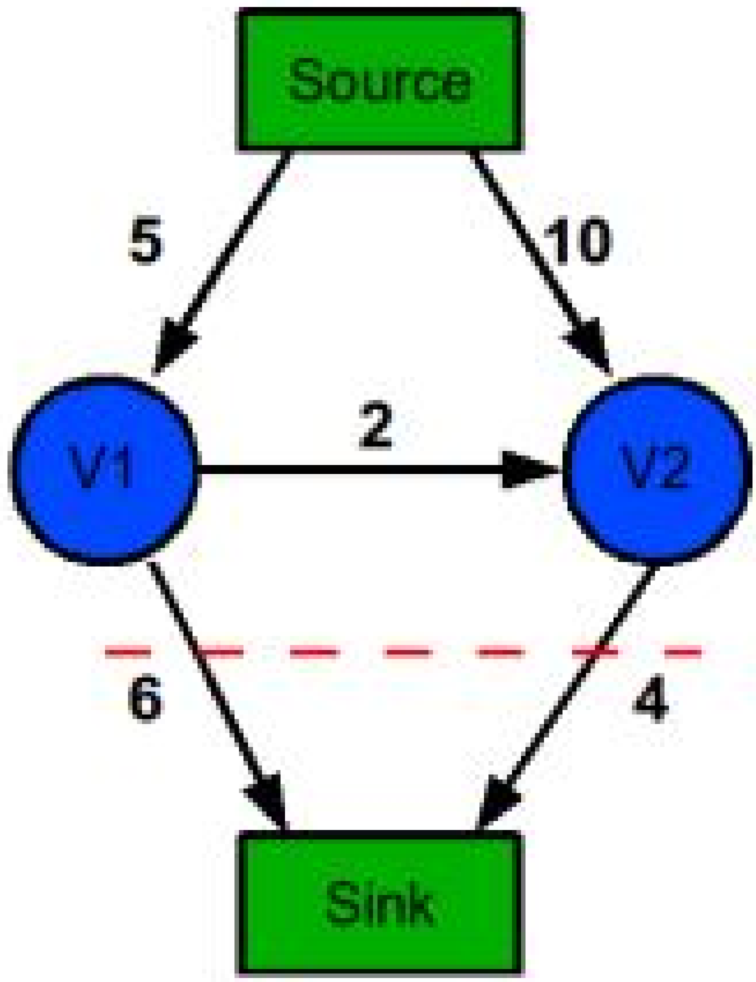

Figure 9.11

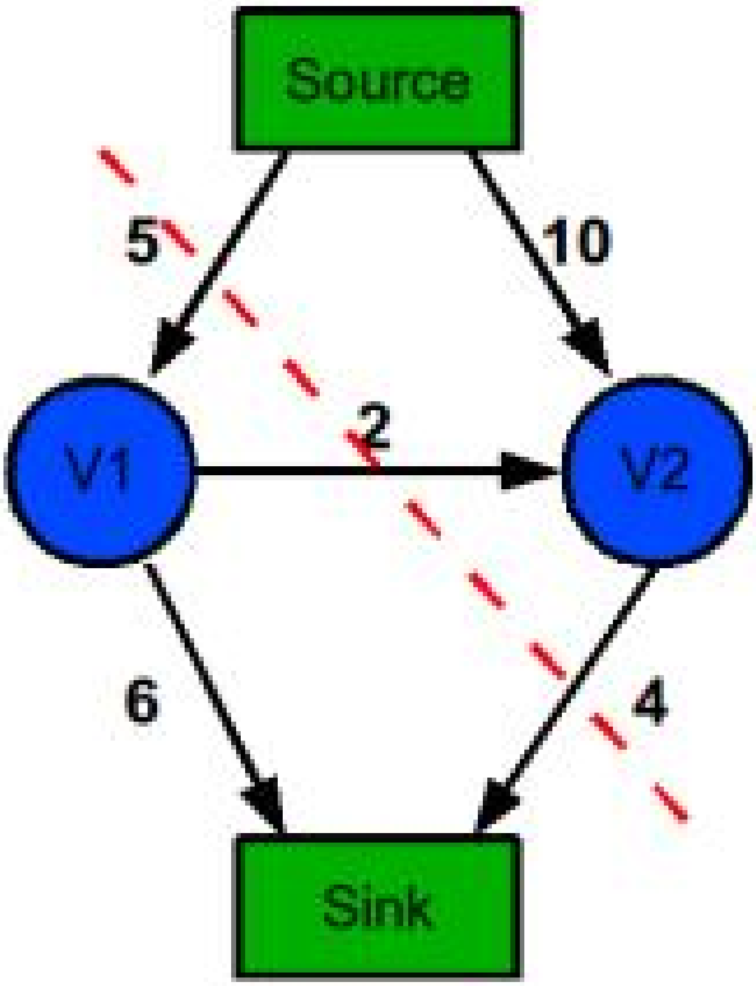

Figure 9.12

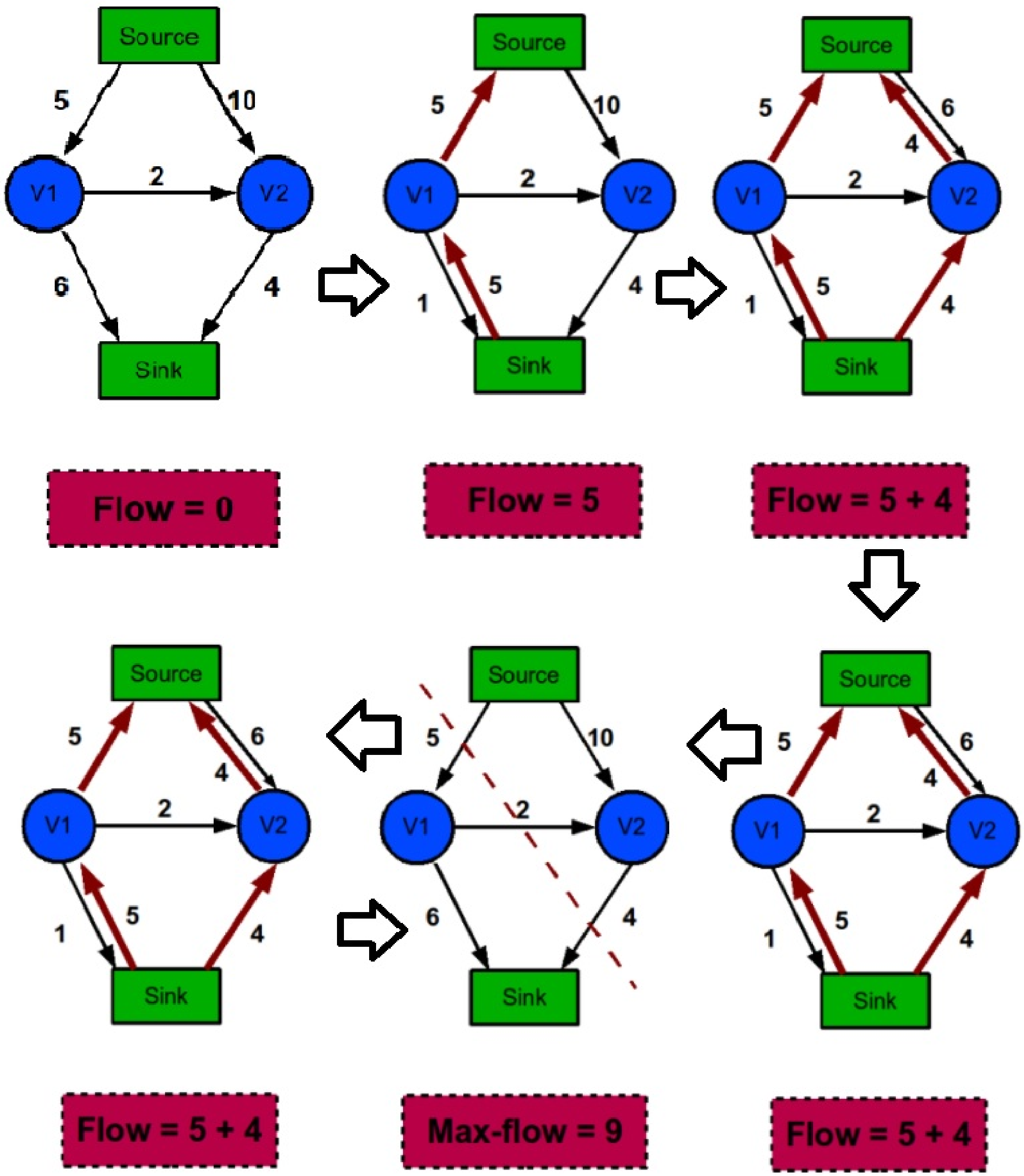

Figure 9.13: Example for Ford-Flukerson Method

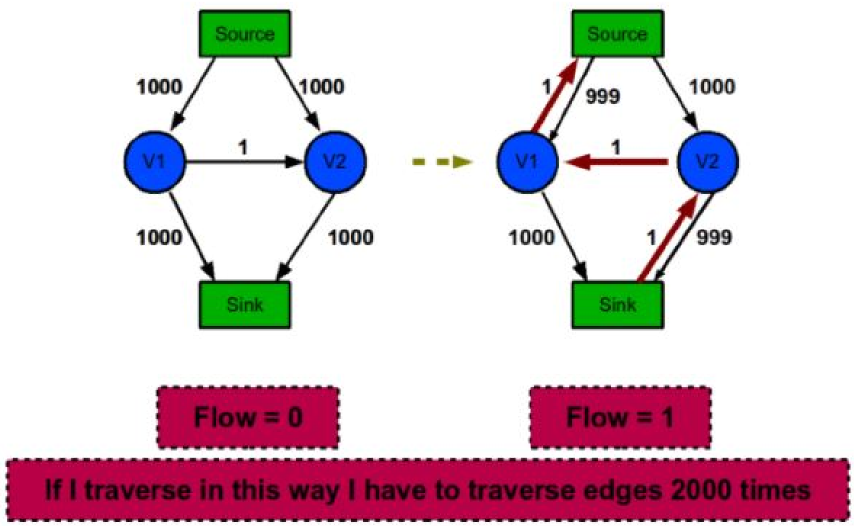
Source
1000
1000
V1
1
V2
1000
1000
Sink
Source
1
999
1000
V1
1
V2
1000
1
999
Sink
Flow = 0
Flow = 1
If I traverse in this way I have to traverse edges 2000 times

Figure 9.14