In this section, we consider single commodity flow problems. The first of these is called the maximum flow problem and its description is given below:

0.0.1 PROBLEM:
Given a directed network \( G = [N; A] \), a special node, \( s \), called the source or origin and a node, \( t \), called the sink or destination and positive numbers \( u_{i,j} \) representing the capacity of arc \((i, j) \in A\) we want to maximize the total shipment from \( s \) to \( t \). This problem is called the maximum flow problem in the literature and is one starting point in this area.

0.0.2 HISTORY:
The special case when all \( u_{i,j} \) are 0/1 was treated by Menger. The general case gives rise to the now famous max-flow-min-cut theorem which seems to have been proved independently by three sets of authors: Elias, Shannon and Feinstein; Ford and Fulkerson; and Robacker. Ford and Fulkerson also gave an algorithm called the labeling algorithm to solve the problem. This algorithm can be shown to different from the simplex algorithm of linear programming in some of its versions. It also preserves the intuitive appeal that is common to this area.

FORMULATION:
Decision Variables:
\( f_{i,j} = \) the flow on arc \((i, j) \in A\)
\( F = \) the total flow across the network. Then:
\[
\sum_j f_{i,j} - \sum_j f_{j,i} = \begin{cases} F & i = s \\ -F & i = t \\ 0 & \text{else} \end{cases}
\]
For example, consider the following example:

The corresponding formulation would be:
Variables: $f_{S,A}, f_{S,B}, f_{A,B}, f_{A,T}$ and $f_{B,T}$. The LP is:

\[
\begin{align*}
f_{S,A} + f_{S,B} &= F \\
f_{A,T} + f_{A,B} &= f_{S,A} \\
f_{B,T} &= f_{S,B} + f_{A,B} \\
f_{A,T} + f_{B,T} &= F \\
0 &\leq f_{S,A} \leq 7 \\
0 &\leq f_{S,B} \leq 3 \\
0 &\leq f_{A,B} \leq 1 \\
0 &\leq f_{A,T} \leq 5 \\
0 &\leq f_{B,T} \leq 6 \\
\text{max } F
\end{align*}
\]

This is, of course, a linear program and can, in principle, be solved by the simplex method. However, the problem could be highly degenerate and unless special precautions are taken it could be very inefficient to use the simplex method. Before describing methods to solve the problem we remark that it is easy to find a feasible solution for this problem as it has been stated. For example, \( F = 0; f_{i,j} = 0 \) for all \((i,j)\) will suffice. First some definitions and a few preliminary results.

A cut separating \( s \) and \( t \) is a partition of the node set \( N \) into two disjoint sets \( S \) and \( \bar{S} \) with \( s \in S \) and \( t \in \bar{S} \).

Let \([F,f]\) be any feasible flow and let \((S,\bar{S})\) be any cut separating \( s \) and \( t \). Then, clearly, \( F \leq u(S,\bar{S}) \), where \( u(S,\bar{S}) = \sum_{i \in S; j \in \bar{S}} u_{i,j} \). \( u(S,\bar{S}) \) is called the capacity of the cut \((S,\bar{S})\). If equality holds then \([F,f]\) is optimal to the maximum flow problem and \((S,\bar{S})\) is a cut whose value is minimum among all cuts separating \( s \) and \( t \).

The following algorithm, known in the literature as the (flow) labeling algorithm, is one way to show that there exist solutions that achieve equality in the relation above. It is usually attributed to Ford and Fulkerson.

**Labeling Algorithm:**

**Input:** A feasible solution \([f,F]\).

**Step 0:** Label \( s (\cdot, \infty) \) and let \( s \in S \), the set of labeled nodes.

**Step 1:** If \( t \in S \) stop; a flow augmenting path has been found. (This is a path along which additional flow can be sent thereby increasing the total flow \( F \).) If not look for a pair \((i,j)\) with \( i \in S \) and \( j \in \bar{S} \) and either (i) \([i,j] \in A \) and \( f_{i,j} < u_{i,j} \) or (ii) \([j,i] \in A \) and \( f_{j,i} > 0 \). If no such \((i,j)\) exists, then stop; \([f,F]\) is the optimal solution to the maximum flow problem and \((S,\bar{S})\) is a minimal cut separating \( s \) and \( t \). If \((i,j)\) of type (i) exists label \( j(i^+, \epsilon_j) \) where \( \epsilon_j = \min \{ \epsilon_i, u_{i,j} - f_{i,j} \} \) and include \( j \) in \( S \). If \((i,j)\) is of type (ii) then label \( i(j^+, \epsilon_j) \) where \( \epsilon_j = \min \{ \epsilon_i, f_{j,i} \} \) and include \( j \) in \( S \).

Return to step 1. Type (i) arcs are called forward arcs and those of type (ii) are called reverse arcs. If we succeed in labeling \( t \), we get an augmenting path as well as the nature of these arcs from the labels themselves. This is done as follows: If the label of \( t \) is \((j^+, \epsilon_i)\) then the previous node to \( t \) in
the flow augmenting path is \( j \) and the last arc is a forward arc; if it is \((j^-, \epsilon_t)\) then the previous node is still \( j \) but the arc is a reverse arc. Now we look at the label of the node \( j \) and the process is repeated until we reach \( s \) and this identifies the entire path. We now augment the flow by \( \epsilon_t \) along the path — by which we mean that flows on forward arcs along the path are increased by \( \epsilon_t \) and those on reverse arcs are decreased by the same amount. This gives us a new feasible flow and the process is repeated. If the starting solution is optimal then at some point before labeling \( t \) we will find no arc of type (i) or (ii). At this point the set \( S \) of labeled nodes gives a minimum cut separating \( s \) and \( t \). Further, all arcs across the cut \((S, \bar{S})\) with the initial end in \( S \) will be saturated \((i.e. f = u)\) and all arcs with the terminal end in \( S \) will be flowless \((f = 0)\). In other words, all forward arcs across the cut will be saturated and all reverse arcs will be flowless. Of course, if this happens with any feasible flow and any cut separating \( s \) and \( t \) then this flow is optimal and the corresponding cut is minimal.

We now illustrate it on the example above:

**Step 2:**

```
\begin{center}
\begin{tikzpicture}
  \node (A) at (2,4) {A};
  \node (S) at (0,2) {S};
  \node (B) at (1,0) {B};
  \node (T) at (3,2) {T};

  \draw[->] (S) -- (A) node[midway,above] {5,3};
  \draw[->] (S) -- (B) node[midway,below] {3};
  \draw[->] (B) -- (T) node[midway,above] {6};
  \draw[->] (A) -- (T) node[midway,above] {5,5};

  \node at (1.5,1) {F=5};
\end{tikzpicture}
\end{center}
```

**Step 3:**
Step 4:

The last is the optimal solution in this example. This was too easy; it did not use the reverse labels. Consider the following example instead: which requires them.

Step 1:
Now we can not send any more flows without backtracking. In other words, the first unit was sent along a "wrong" route. If we could take it back, then we can send two units of flow thereby increasing the total by one unit. Thus, we wish to take back the unit flow along S-A-B-T, then add one unit flow along each of the routes S-A-T and S-B-T. The net effect of all this on the flows on each arc is as follows:

SA: one unit decrease followed by one unit increase — net result no change
SB: one unit increase
AB: one unit decrease
AT: one unit increase
BT: one unit decrease followed by one unit increase — net result no change

We can summarize all this by viewing this as one unit flow "along" the "path" S-B-A-T where this means that we increase on arcs SB and AT which are called forward arcs with respect to this path since their directions agree with that of the path and decrease on the arc AB a reverse arc since its direction is opposite. This is what the reverse labels do for us. The next picture is after this is done is:
On this final picture, if we do the labeling we get the min cut diagram illustrated below: