5.3 Maximum Flow Problem

This problem involves a directed network with arcs carrying flow. The only relevant parameter is the upper bound on arc flow, called *arc capacity*. The problem is to find the maximum flow that can be sent through the arcs of the network from some specified node *s*, called the source, to a second specified node *t*, called the sink. Applications of this problem include finding the maximum flow of orders through a job shop, the maximum flow of water through a storm sewer system, and the maximum flow of product through a product distribution system, among others.

A specific instance of the problem with source node *s* = A and sink node *t* = F is shown in Fig. 15 with its solution. In particular, the solution is the assignment of flows to arcs. For feasibility, conservation of flow is required at each node except the source and sink, and each arc flow must be less than or equal to its capacity. The solution indicates that the maximum flow from A to F is 15.

A cut is a set of arcs whose removal will interrupt all paths from the source to the sink. The capacity of a cut is the sum of the arc capacities of the set. The *minimal cut* is the cut with the smallest capacity. Given a solution to the maximum flow problem, one can always determine that at least one minimal cut, as illustrated in Fig. 16. The minimal cut is a set of arcs that limit the value of the maximum flow. Not coincidentally, the example shows that the total capacity of the arcs in the minimal cut equals the value of the maximum flow (this result is called the max-flow min-cut theorem). The algorithm described in this section solves both the maximum flow and minimal cut problems.

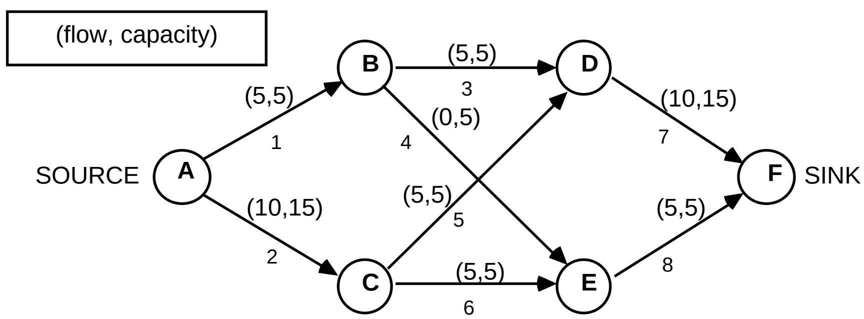

Figure 15. Example maximum flow problem with solution

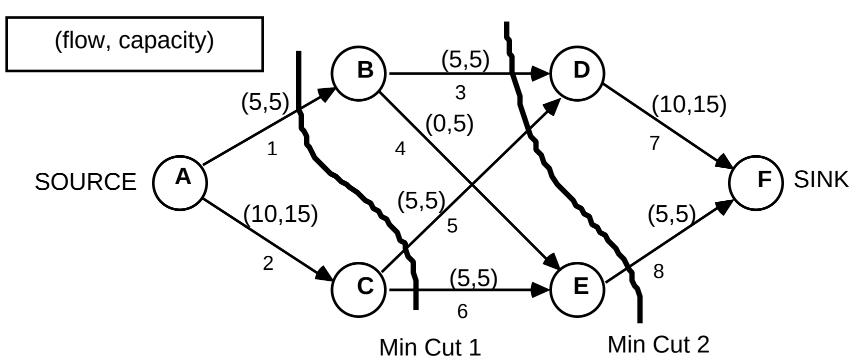

Figure 16. Minimal cuts determined by the maximum flow

Flow Augmenting Algorithm

The traditional way to solve the maximum flow problem is with the flow augmenting algorithm developed by Ford and Fulkerson (). The algorithm begins with a feasible set of arc flows obtaining some value, v_0, for the flow out of the source and into the sink. A search is then made in the network for a path from source to sink that can deliver an increased flow. This is called a flow augmenting path. The flow is increased along that path as much as possible. The process continues until no such path can be found, at which time the algorithm terminates.

Before stating the algorithm formally, we illustrate the general idea with the example. In the presentation, let x_k be the flow on arc k. For an initial flow solution we assign 0 flows to all arcs and $v_0 = 0$, as shown in Fig. 17.

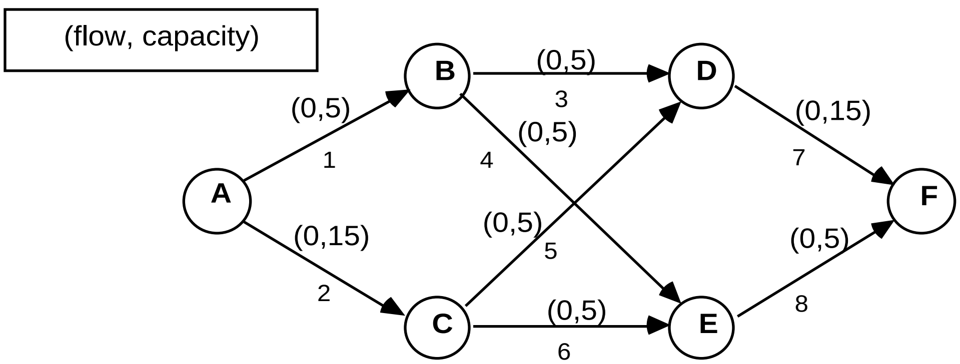

Figure 17. Initial flow with $v_0 = 0$

Flow may be increased in an arc when the current flow is less than capacity ($x_k < u_k$). Flow can be decreased in arc when the current flow is greater than zero ($x_k > 0$). As we will see, a flow augmenting path traverses an arc in the forward direction when the arc flow is to be increased and in the reverse direction when the arc flow is to be decreased. For the initial solution there are a number of flow augmenting paths. We choose the path $\boldsymbol{P}_1 = (1, 4, 8)$, identified by the list of arcs in the path, and note that the flow may be increased by 5 to obtain the solution of Fig. 18.

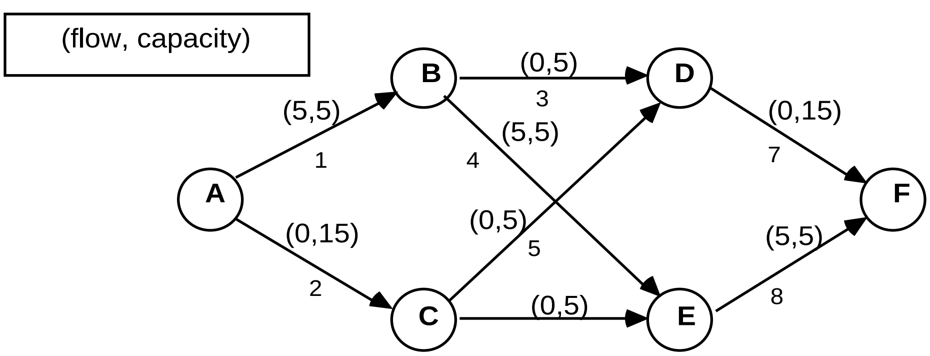

Figure 18. Augmenting flow on (1, 4, 8) giving $v_1 = 5$

By observation we discover another flow augmenting path in Fig. 18, $\boldsymbol{P}_2$ = (2, 5, 7). Augmenting the flow on this path by 5 we obtain the solution in Fig. 19.

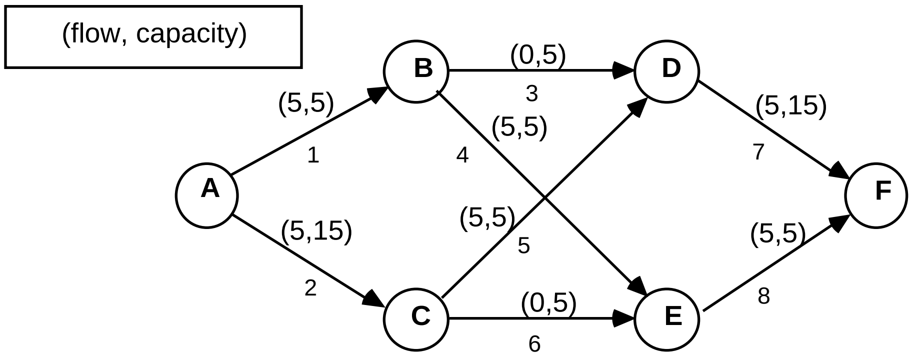

Figure 19. Augmenting flow on (2, 5, 7) giving $v_2 = 10$

From Fig. 19 we discover a flow augmenting path of a different nature than the two previously described. Searching from A we find that flow can be increased in arcs 2 and 6, but that further progress is blocked in arc 8. We can however increase the flow from node E to B by decreasing the flow in arc 4 since its flow is greater than zero. Finally flow can be increased in arcs 3 and 7 because the flows in these arcs are less than arc capacities. The path describing this sequence of flow changes is $\boldsymbol{P}_3$ = (2, 6, – 4, 3, 7), where the positive numbers show arcs in which flow is to be increased and negative numbers show arcs in which flow is to be decreased. The flow can be increased by 5 along this path to obtain the results of Fig. 20.

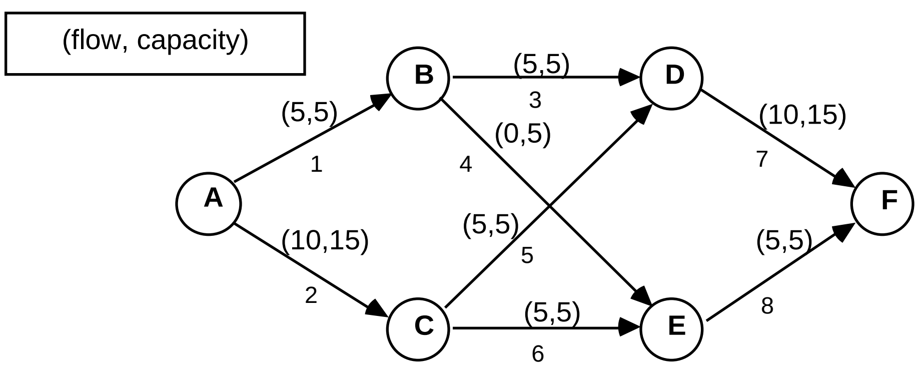

Figure 20. Augmenting flow on (2, 6, -4, 3, 7) giving $v_3 = 15$

Review of Fig. 20 reveals no additional flow augmenting paths so the maximum flow has been obtained. To find the minimal cut, we identify a set of nodes, call it $\boldsymbol{S}$, such that one or more paths exist from the source node to the nodes in $\boldsymbol{S}$ on which additional flow can be delivered. The source node is necessarily in $\boldsymbol{S}$. The set of arcs leaving $\boldsymbol{S}$ comprises the minimal cut. For example, starting at node A, we can reach only node C through an arc that has additional capacity. Thus we have the node set $\boldsymbol{S} = \{A, C\}$ to which additional flow could be advanced. The remaining nodes define the set $\bar{\boldsymbol{S}} = \{B, D, E, F\}$. The minimal cut consists of all the arcs that pass from $\boldsymbol{S}$ to $\bar{\boldsymbol{S}}$, arcs $\boldsymbol{C} = \{1, 5, 6\}$ in the example. The flows in the arcs in the minimal cut are at their respective arc capacities because the cut is the bottleneck for the total flow. Alternatively, the flows in arcs that pass from $\bar{\boldsymbol{S}}$ to $\boldsymbol{S}$ are at zero.

Formal Algorithm

For this algorithm we identify the source as node s and the sink as node t. An initial feasible flow solution is assumed available for the algorithm with flow into node t (and out of node s) equal to v_0.

Do the following until no flow augmenting paths may be found.

a. Find a flow augmenting path defined by the sequence of arcs $\boldsymbol{P} = (k_1, k_2, \ldots k_p)$ where p is the number of arcs in the path. (An arc will have a positive sign if it is traversed in the forward direction and a negative sign if it is traversed in the reverse direction.)

b. Determine the maximum flow increase in the path.

$$\delta = \min \begin{cases} \min (u_k - x_k : k > 0, k \in \boldsymbol{P}) \\ \min (x_{-k} : k < 0, k \in \boldsymbol{P}) \end{cases}$$

c. Change the flow in the arcs on the path. Let x'_k be the new value of flow. For every arc k on the path $\boldsymbol{P}$

$$x'_k = x_k + \delta \text{ if } k > 0 \text{ and } x'_{-k} = x_{-k} - \delta \text{ if } k < 0.$$

This algorithm does not derive from the simplex algorithm and does not maintain a basic solution. It is a representative of a large class of non-simplex algorithms for network flow problems.

Finding Flow Augmenting Paths

In the example problem, flow augmenting paths were discovered by observation. For larger networks and for computer implementation, a more formal procedure is required. The algorithm described below is a search procedure that begins at s and labels all nodes to which a flow augmenting path from s can be found. When node t is labeled, we have a *breakthrough*, and the required flow augmenting path to t has been discovered. When the algorithm begins, all nodes are unlabeled except s. We "check" a labeled node after all avenues for finding paths from the node have been explored. Initially, all nodes are unchecked.

Labeling Algorithm

Do the following until node t is labeled or there exists no labeled unchecked node.

a. Select a node i that is labeled but unchecked. For each arc $k(i, j)$ originating at node i such that node j is unlabeled and $x_k < u_k$, label node j with the index of the arc k. For each arc $k(j, i)$ terminating at node i such that node j is unlabeled and $x_k > 0$, label node j with the negative index, $-k$.

b. Check node i.

If the algorithm terminates with t labeled, the flow augmenting path is found by tracing the path backward from t and constructing ***P*** from the labels encountered. If the algorithm terminates with t unlabeled, no flow augmenting path exists. We illustrate the former case with the situation of Fig. 19. The resultant node labels and checks are shown adjacent to the nodes in Fig. 21. An x indicates that a node is checked.

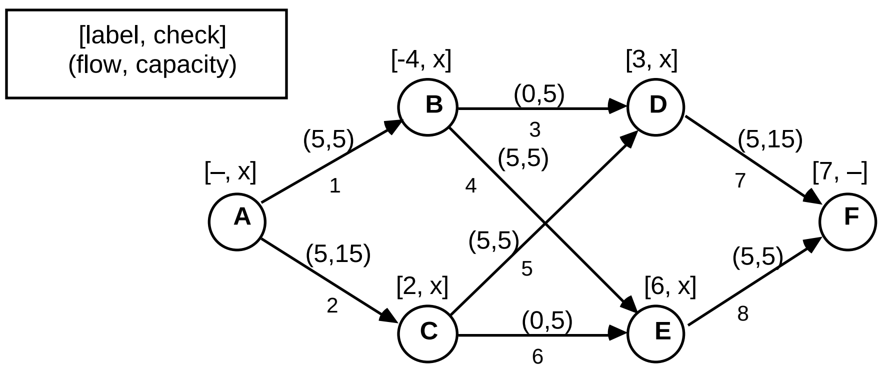

Figure 21. Illustration of labeling algorithm

When node F is labeled, the flow augmenting path is complete. The label on node F indicates the last arc on the path is 7. Node D is the origin of this arc, so we find the next arc on the path from the label on node D. The process continues until the path ***P*** = (2, 6, -4, 3, 7) is discovered.

The sink node is not labeled when we apply the algorithm to the situation described in Fig. 20. The labels and checks for this case are shown adjacent to the nodes in Fig. 22.

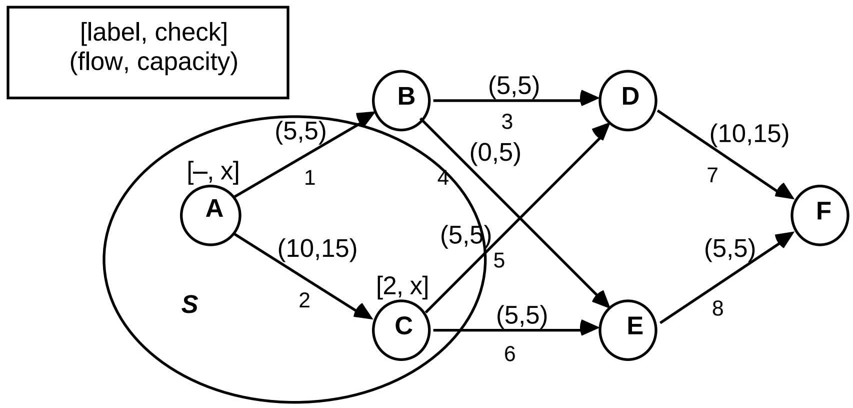

Figure 22. The minimal cut discovered by the labeling algorithm

Although we fail to find the flow augmenting path, we do identify the minimal cut. ***S*** is the set of nodes labeled in this final attempt. The arcs in the minimal cut are those that leave the set, in this case the cut is ***C*** = {1, 5, 6}. Adding the capacities of these arcs we obtain the value 15, the same as the value of the maximum flow from *s* to *t*.