

## Transportation Planning: The maximum flow problem

In many applications, and especially in the planning of Transportation and Logistics of materials, one important problem is that of finding *how much material can be shipped* between two locations, using *transportation channels of limited capacity*. Different variations of this problem are faced by the shipping industry, by companies that must maintain a certain amount of inventory in different warehouses, etc. Another application is in the design of transportation routes (e.g. the Kowloon Motor Bus company may want to know how the maximum number of people that can be transported from Choi Hung to Clear Water Bay Beach on a day.)

Let us look at some example problems of this type before we study a technique to solve such problems.

### Example 1.

The HK government plans to hold a Super Light-and-Sound Show in the Victoria Harbor on New Year eve. The organizing committee has asked the HKE power company about the maximum electric power that can be supplied at the WanChai Sub-Station for use in the celebration events. The power generator for this is located on Lamma Island. It will be supplied to the WanChai station for utilization for the event. The following figure shows the maximum available capacity (after subtracting other normal use requirements) for transferring power between different sub-stations on HK Island. What is the maximum power that HKE can promise to the event organizer?

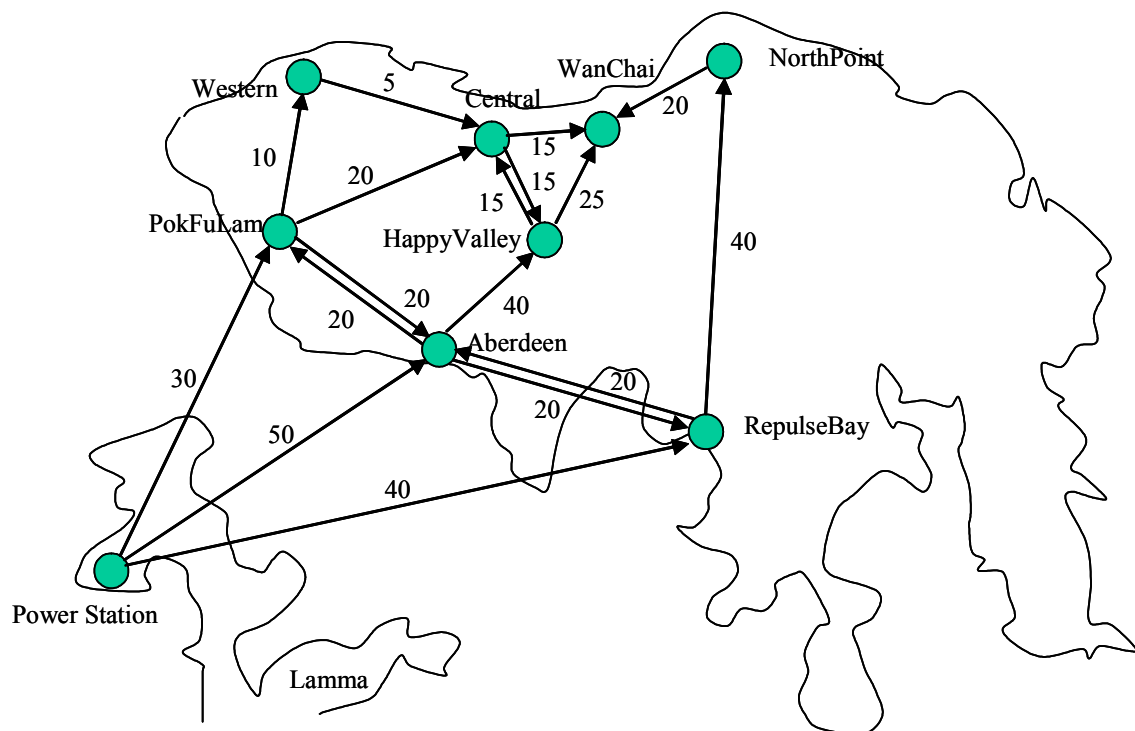


Figure 1. Electric power transmission capacities for example 1

### Example 2.

A company has a factory located in Detroit, USA, producing electric golf carts. The carts will be exported to Asian countries, and are therefore stored in a warehouse in San Francisco. All carts are shipped from the company to San Francisco by train. The train company operates trains daily on the national rail network. The carts are packed in train-compartments, called bogeys. There is a limit on how many compartments can be shipped across each link of the train network daily. The network below shows the carrying capacity. What is the maximum number of bogeys that can be shipped out of Detroit each day?

[Note that although some routes take longer to reach the destination than others, this does not matter to our problem – in steady state, each day we ship out some bogeys full of carts, and each day, we receive the same number of bogeys full of carts at San Francisco].

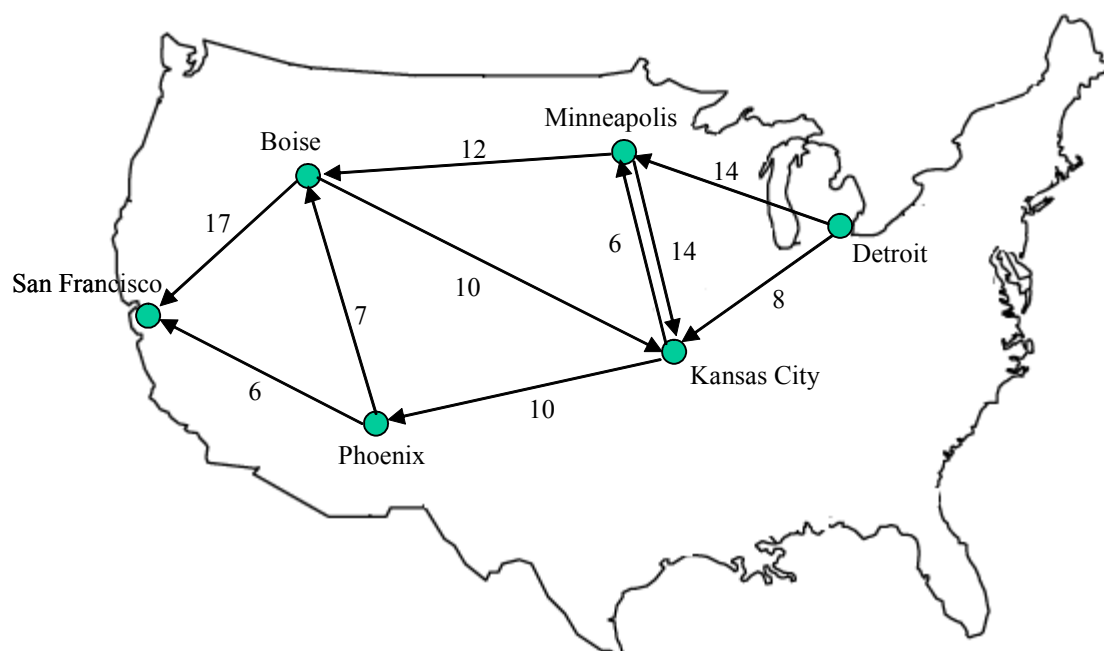


Figure 2. Golf cart shipping example

It is conventional in these problems that the location which is producing the goods to be sent out is called a **source**, while the location of the receiving point is called the **sink**. In both our examples, we have assumed that the source and sink have infinite capacity. This is convenient in solving the problem.

An important consideration for all maximum flow problems is that there is no leakage, or production, of flow at any node except for the source and sink.

You must have noticed that the two examples are really very similar in nature, and each tries to **maximize the flow from source to sink** across a **network** of **limited capacity channels**. This type of problem has a very natural way to model it, using graphs. Such problems can be modeled using a technique of algebra called Linear Programming, and therefore it is possible to use the Simplex method to solve them (also taught in this course). However, we shall study a graph theory method, which is more efficient.

## The Ford-Fulkerson Max Flow Algorithm for Networks

We shall represent the network as a graph, where the weight of each edge will denote the maximum capacity of that link of the network. To make things easier, we first look at some basic ideas used by the algorithm.

**Flow Cancellation:** Consider two adjacent nodes with a positive capacity in both directions, as shown in the figure below. Consider a case where 5 units are transported from node  $a$  to node  $b$ , and 3 units from node  $b$  to node  $a$ . The net result is identical to a situation where 2 units are shipped from node  $a$  to node  $b$ , and 0 units from node  $b$  to node  $a$ . We may say that the flow in one direction cancels the flow in the reverse direction.

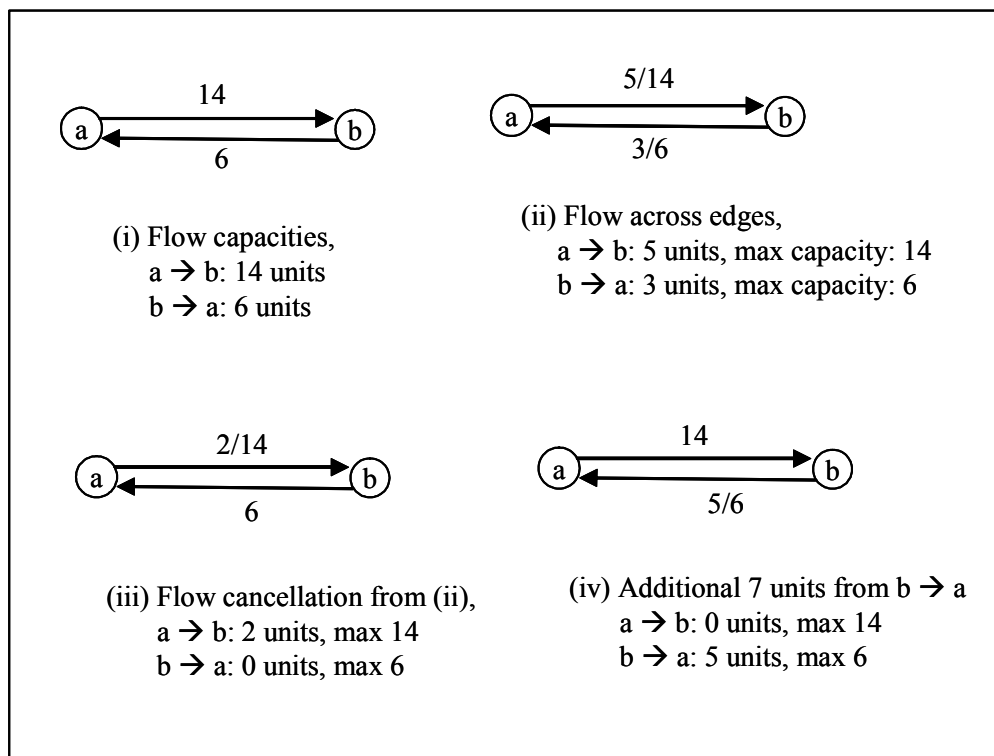


Figure 3. Flow cancellation

Using flow cancellation, we can always reduce the material flow between a pair of adjacent nodes to a positive flow on only one of the edges, with zero flow on the other edge. The reason it can always be applied is that (a) it obeys the law of conservation of mass at each node, and (b) since it only reduces the actual flows in each edge, it never leads to a flow capacity constraint violation.

Flow cancellation is a simple and useful trick we shall use to solve the maximum flow problem. In particular, it helps us to clearly see some possible flows which “appear” to be infeasible, but are actually “feasible”. Consider the example in Figure 3 above. The maximum capacity of edge  $a \rightarrow b$  is 14 units, while maximum capacity of edge  $b \rightarrow a$  is 6 units. A flow of 5 units from  $a$  to  $b$ , and flow of 3 units from  $b$  to  $a$  is identical to an actual flow of 2 units from  $a \rightarrow b$  and no flow from  $b \rightarrow a$ . Assume that we try to send an additional 7 units from  $b \rightarrow a$ . By just looking at the capacity of the edge from  $b \rightarrow a$ , it seems that this cannot be done, since the flow exceeds the max capacity of this edge,

which is 6 units. However, after flow cancellation, we get an equivalent situation as shown in Figure 3 (iv), which is clearly feasible.

**Residual Network:** Our approach for solving the maximum flow problem will be to find, at each step, a path that can carry some amount of flow from the source to the sink. Obviously, once we commit to some flow, the edges along this path will have less *remaining capacity*. To assist in our search for the next flow path, it is convenient to keep track of the remaining capacity in each edge. The residual network is a graph that indicates the remaining net capacity for each edge.

In constructing residual networks, the following convention is convenient. If two adjacent nodes only have a single, directed edge between them, then we add another edge with an initial capacity = 0, in the other direction. The reason for this will become clear when we construct a residual network.

We shall solve our problem in stages, adding some more flow from source to sink in each stage. A residual network of the graph will be maintained at each stage. This helps us in (a) keeping track of the remaining capacity to carry extra material on each edge, and (b) to search for the next path over which we can send more materials.

We look at an example of how to maintain residual networks, using the Golf cart shipping example above. Figure 4a shows the initial situation, where the flow is zero; the cities represent the nodes which are named by the first letter of the city. Figure 4b shows the same graph with the 0-capacity nodes added. Next assume that we send 6 units along the path  $\langle D, M, K, P, S \rangle$ , as shown in Figure 4c. This gives us the residual network as shown in Figure 4d. Consider the edge  $D \rightarrow M$  in the initial graph, with capacity 8. Since 8 units flow from D to M, so its residual capacity is 6 units. In reality, there is no edge from  $(M \rightarrow D)$  in the graph; but we represent this as an edge of zero capacity. Sending 6 units from  $D \rightarrow M$  can be seen as increasing our ability to send up to 6 units from  $M \rightarrow D$ . This does not really mean that we can send some units on  $M \rightarrow D$ , it just means that if we actually used some of the capacity ( $\leq 6$  units) of this fake-edge,  $M \rightarrow D$ , then that flow could just be **cancelled** by the 6 units we are sending along  $D \rightarrow M$ . The net result will be a situation that does not violate our initial capacity constraints.

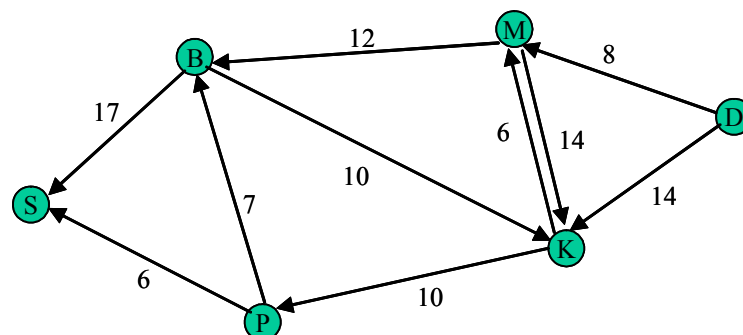


Figure 4 (a). The flow network for golf cart shipments

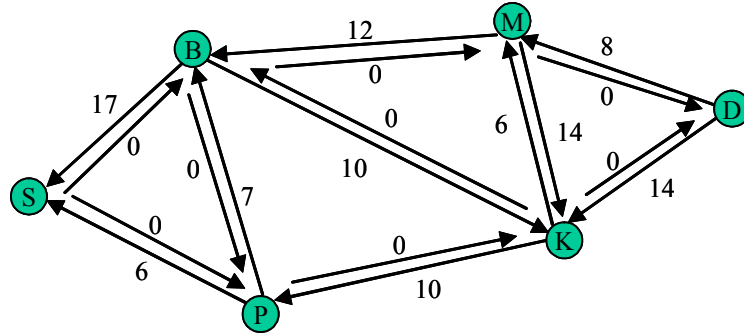


Figure 4 (b). The equivalent flow network (0 capacity edges added)

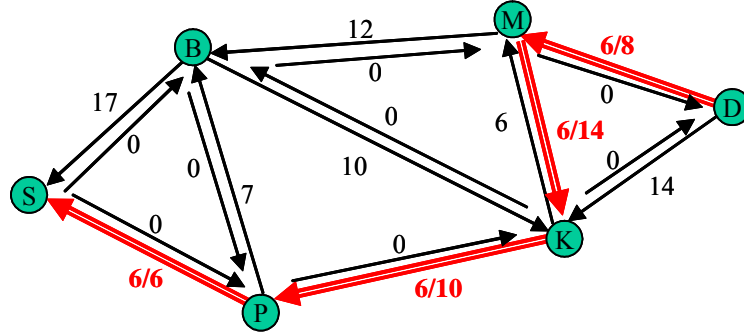


Figure 4 (c). A flow of 6 units along  $\langle D, M, K, P, S \rangle$

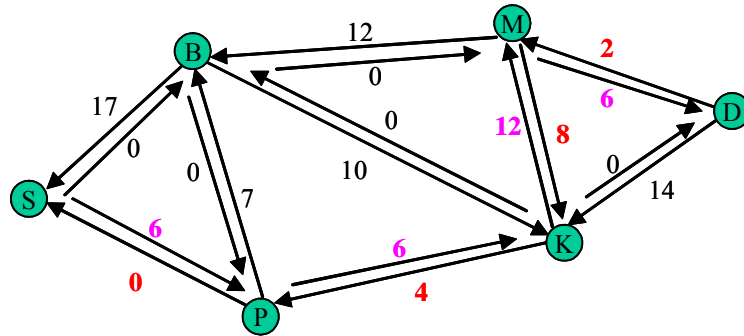


Figure 4 (d). The residual network of  $G$  after accounting for the flow in (c)

The following property of the residual network is very useful:

**Property 1.** Let  $G$  be a network with zero flow. Let  $G_{f1}$  be its residual network after introducing a flow,  $f1$ , from the source to the sink, of  $|f1|$  units.

Assume that we can now introduce a flow,  $f2$ , of  $|f2|$  units in the residual network  $G_{f1}$ , without violating any capacity constraints, resulting in a new residual network  $G_{f1,f2}$ .

Then the new residual network is identical to the one we would obtain if we had introduced a flow of  $|f1| + |f2|$  units directly into the network  $G$ .

**Augmenting Path:** Any simple path in a residual network, going from the source to the sink, such that each edge on this path has capacity  $> 0$ , is an augmenting path.

By “simple path”, we mean that it should not cross itself, i.e. no node should occur twice in a simple path.

An augmenting path is just some path that allows us to send more units across our residual network.

The above definitions are now sufficient for us to write the Ford-Fulkerson method:

---

***The Ford-Fulkerson Method for finding the maximum flow in a given network:***

Step 1. Initialize the network with a zero flow

Step 2. If we can find an augmenting path,  $p$ , in the network, then

2.1. Apply the maximum flow allowed on  $p$  to the network;

2.2. Compute the residual network;

2.3. Return back to Step 2;

otherwise: the current residual network is carrying the maximum flow.

---

In short, if we reach a situation where there is no augmenting path in the residual network, then the total flow in this state is the maximum flow that the network could allow.

The beauty of the method is that at each step, we just find any way to improve the total amount of flow, and we will eventually reach the optimum. Another way to look at this is that our search for the optimum solution is along a monotonically increasing path.

**Proof of correctness**

How do we know that the solution we have reached is optimal? Consider this: at some intermediate stage, if we had selected a different augmentation path, then certainly we would have obtained a different residual network. Ford-Fulkerson claim that no matter what intermediate choices we make, when there are no more augmentation paths available, *the flow in the corresponding residual networks will be the same !*

To prove this, we need to re-use our old trick of putting a part of the network in a bag. In this case, we construct a “bag” that includes the source and possibly some other nodes; the sink, however, must stay outside the bag. Figure 5 shows an example, with the red nodes inside the bag, and the green nodes outside it.

**Property 2.** At some stage, let the flow from source to sink be  $f$ , carrying a total amount  $|f|$  from source to sink. Then, for *every bag* constructed as described above, the *net amount* flowing out of the bag is  $|f|$ .

Why is this true? Assume it was false, and that the net flow was some amount,  $x < |f|$ . Now the source is sending out  $|f|$  amount each day, and only  $x$  units are flowing out every day. Thus some node (or set of nodes) inside the bag must be saving up the difference  $= (x - |f|)$ . If we use the same network for a long time, each day we will build up some extra amount inside the bag, eventually carrying a large store inside the bag. However, this obviously contradicts our assumption that the net flow at each node (*total inflow – total outflow*) must be exactly zero. The same logic proves that  $x > |f|$  is impossible.

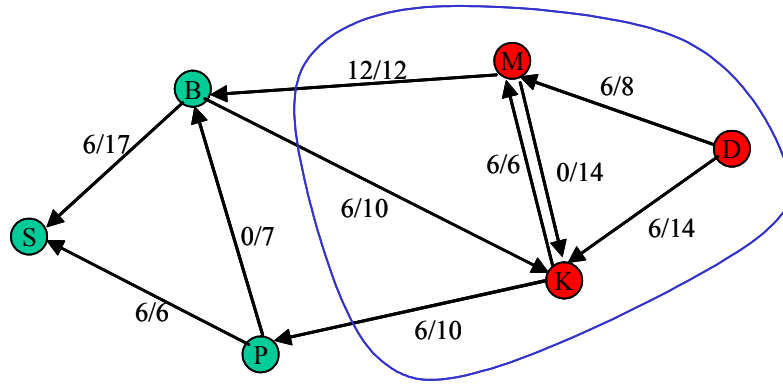


Figure 5. A source-containing “bag” in the network for our example  
net flow =  $12 - 6 + 6 = 12$  units; outflow capacity =  $12 + 10 = 22$  units

To complete our proof of correctness, we need a definition: the **outflow capacity of a bag**. The outflow capacity of a bag is the maximum amount that can *flow out* of the bag. Notice that when compute the outflow capacity, we just ignore the amount that can flow into the bag, because we are interested in the maximum amount that can flow out of the bag – this will happen when we are sending out the maximum, and taking in zero. The outflow capacity for the bag in Figure 5 is therefore 22 units, corresponding to the edges  $\{(M,B), (K,P)\}$ .

Assume that  $f$  is a flow after which our method can find no augmenting path from source to sink in graph  $G$ . At this point, we partition all nodes of  $G$  into two sets: any node to which we can still find an augmenting path is placed in the set IN-BAG; all other nodes are placed in the set OUT-OF-BAG. Clearly, the node sink is in OUT-OF-BAG.

The *outflow capacity* of IN-BAG = the volume of flow  $f = |f|$ .

The reason for this becomes clear if we generate the residual network for flow  $f$  through  $G$ . In the residual network, every edge going from IN-BAG  $\rightarrow$  OUT-OF-BAG must have zero residual capacity. If not, then of course we could push some more flow across this edge, and therefore the corresponding vertex should not have belonged to OUT-OF-BAG.

Let us focus our attention on this partitioning of nodes. If there is some flow that can push a larger amount than  $|f|$ , say  $f^+$  through our network. Then, from property 2, the flow going out of IN-BAG must also be  $f^+$ . But this is clearly impossible, since the maximum outflow capacity of the edges going out of IN-BAG is limited to  $|f| < f^+$ .

This completes the proof that the Ford-Fulkerson method must have terminated at the optimum, i.e. maximum flow that can be sent through the network.

Before completing this section, I would like to point out a few things of interest.

1. In classical terminology, the notion of the “bag” that partitions our set of nodes into two non-intersecting subsets is called a “cut” of the graph. The proof of correctness of the max-flow method above was completed by finding a particular cut that had the minimum outflow capacity. Therefore this method is sometimes called the **max-flow min-cut** method. Stated in the classical terminology, we say that the maximum flow we can send across a network equals the minimum of the outflow capacities of all possible cuts of the graph (as long as the cut separates the source from the sink, of course).

2. The problem of finding maximum flow capacity of networks has many applications in Modern Logistics management. Computer programs that can solve max-flow programs are used heavily by all automatic systems that calculate how shipments are transported by shipping companies (they may use planes, ships, trains, etc.).

3. Another common use of max flow problems is in optimization of design. This includes design of piping systems for chemical and food-processing plants, water supply of a city, sewage systems, etc. One difference in the design of water supply systems is that usually there is one source (the reservoir), and many sinks (e.g. the water storage tank of each area, from where it is pumped to individual apartments). After completing the design of a particular supply network, we can then compute the maximum amount of water that can be supplied to any of the supply points. If this amount does not meet our required demand in that area, we will know that a higher capacity pipe should be designed.