Maximum Flow

Now we address the final question of the Seervada Park management: how to route the various tram trips from the entrance to the waterfalls to maximize the number of trips per day. There are strict upper bounds on the number of outgoing trips per day (we will assume that trams return by the same route, so we can concentrate only on the outgoing trips) on each arc. The limits are represented in figure 3. In general, the limit (or *capacity*) of arc (i,j) will be denoted by c_{ij}.

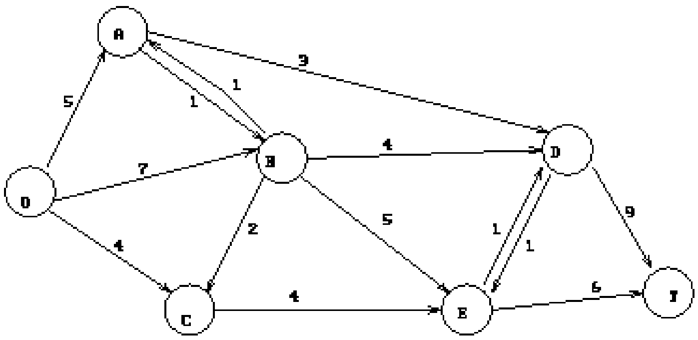

Figure 3: Daily limits

Note that the arcs are directed and it is possible to have oppositely directed arcs between the same nodes. Given the limits, one possible solution is to send twelve trams per day, with five using the route O-B-E-T, one using O-C-E-T, one using O-C-E-D-T, three more on O-A-D-T and two on O-B-D-T. Is there a way to do better?

A very efficient algorithm for this problem is the *augmenting path algorithm*, due to Ford and Fulkerson. To describe this algorithm, it will be useful to introduce the following notation. The traffic (or *flow*) on arc (i,j) is denoted by f_{ij}. For example, in the twelve tram solution described above, $f_{OA} = 3$ (3 trams use arc **OA**), $f_{OB} = 5+2 = 7$ (7 trams use arc **OB**, namely 5 from route O-B-E-T and two from O-B-D-T), $f_{OC} = 1+1 = 2$ and so forth.

An *augmenting path* is a undirected path from the origin to the destination such that every arc directed in the forward direction of the path has positive *residual capacity* $c_{ij} - f_{ij}$ and every arc directed in the backward direction of the path has positive flow f_{ij}. Each augmenting path represents an opportunity to increase the amount of flow sent from the origin to the destination. Note that such a path may include both arcs that represent increasing flow (the forward arcs) and arcs that represent cancelling flow (the backward arcs). The maximum amount of flow that can be feasibly sent along the path is determined by the smallest of the residual capacities along forward arcs *and* of the flows along backward arcs.

This gives the augmenting path algorithm:

1. Identify an augmenting path from the origin to the destination.
2. Find the maximum amount of flow that can be feasibly sent along this augmenting path. Let this value be **m**.
3. Increase the flow on forward arcs of this path by **m** and reduce the flow on backward arcs by **m**.

Note that step 1 is not completely defined. There are many alternatives for finding an augmenting path.

When we solve the example, the paths will be chosen arbitrarily (and you may also choose them arbitrarily in the homework and on exams). Note, however, that you must be organized enough to always find an augmenting path when one exists. So it is in fact a good idea to apply a labeling procedure (such as Ford and Fulkerson's method, described in many textbooks) when you cannot find more augmenting paths by inspection.

Let's work through our example: is the twelve tram solution described above an optimal solution?

By inspection, we discover that the path O-A-B-D-T, which only contains forward arcs, has positive residual capacity on every arc. Namely, the residual capacity on arc **OA** is 2, it is 1 on **AB**, 2 on **BD** and 3 on **DT**. The smallest of these values is 1. Therefore we have found an augmenting path and we can obtain a better solution by sending one unit of flow along this path. This yields a thirteen tram solution. Is it optimal?

Again, by inspection, we discover the following augmenting path: O-C-E-B-D-T. It contains four forward arcs and one backward arc (**BE**). The residual capacities on the forward arcs are 2, 2, 1 and 2 respectively. The flow on the backward arc is 5. The smallest of these values is 1. Therefore we can increase our total flow by 1 (to a total of fourteen trams). This is acheived by increasing the flow on each of the forward arcs by one unit and decreasing it by one unit on the backward arc. The resulting arc flows are shown in figure 4.

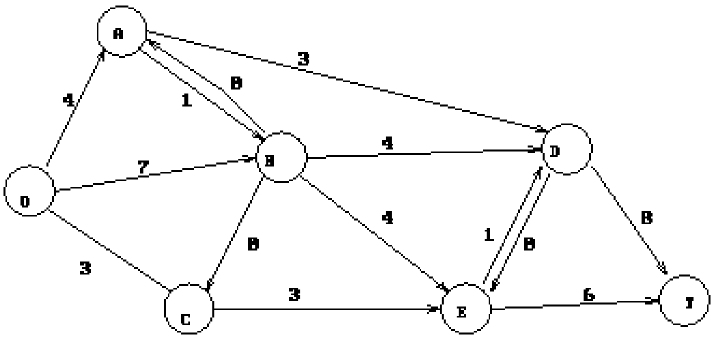

Figure 4: Optimal flow

Note that routes for the trams can easily be constructed from the flow solution. For example, three trams can use route O-C-E-T, three O-B-E-T, one O-B-E-D-T, three O-B-D-T, three O-A-D-T and one O-A-B-D-T. There are many other ways of routing the fourteen trams. We are happy with just one. Can we do better than fourteen trams?

We apply the Ford-Fulkerson labeling procedure. The origin **O** is labeled. Then **A** is labeled since there is a positive residual capacity on arc **OA**. Similarly, **C** can be labeled and then **E** (since **OC** and **CE** have positive residual capacities). Next **B** can be labeled since **BE** has positive flow and **E** is already labeled. Now all arcs going from a labeled node to an unlabeled one have zero residual capacity and all arcs going from an unlabeled node to a labeled one have zero flow. So no further labeling is possible. Since the destination could not be labeled, there is no augmenting path to be found. The current solution is optimum.

There is a simple way to see that, when the destination cannot be labeled by the labeling procedure, we have found an optimum flow: look at the arcs going out of the labeled nodes, namely O-A-B-C-E. The total capacity of the arcs leaving this set is 14. Therefore, we know that we cannot send more than 14 units of flow from **O** to **T**. Since we found a way to send 14 units, we know we have the maximum flow. This is not a coincidence. Call a **cut** a set of arcs leaving a node set that contains the origin and doesn't contain the destination. Let the capacity of a cut be the sum of the capacities of arcs in the cut. The **max-flow min-cut theorem** states that the maximum feasible flow from the origin to the destination equals the capacity of a cut with minimum capacity. To prove this theorem, we could write the maximum flow problem as a linear program (we will actually do this in the minimum cost flow section) and observe that the dual of this linear program is the minimum cut problem. The max flow- min cut theorem then follows from the strong duality theorem of linear programming. (The only tricky part of the proof is to show that the max-flow and the min-cut problems can indeed be formulated as linear programs instead of integer programs. The key property here is known as *(total) unimodularity*. We will not go into it in this course.

Exercise 4 *Consider a network with nodes s, a, b, c, d, t where s is the origin, t is the destination, and arcs (s,a), (s,b), (s,c), (a,b), (a,d), (b,c), (b,t), (c,t), (d,b), (d,t). The following table contains arc capacities as well as arc flows.*

	Arcs									
	(s,a)	*(s,b)*	*(s,c)*	*(a,b)*	*(a,d)*	*(b,c)*	*(b,t)*	*(c,t)*	*(d,b)*	*(d,t)*
Capacities	10	4	6	2	6	2	7	4	3	6
Flows	8	4	2	2	6	2	7	4	3	3

(a) Verify that the solution given by the above arc flows is feasible, that is it satisfies the capacity restrictions and there is conservation of flow at each node of the network other than the origin and destination. (Conservation of flow at node a means that the total flow going into node a equals the total flow leaving it.)

(b) Is the above solution a maximum flow from s to t ? If not, find an augmenting path and construct a maximum flow.

(c) Find a cut of minimum capacity.

Michael A. Trick
Wed Nov 8 14:14:52 EST 1995