Shortest Paths

Consider an *undirected* and *connected* network with two special nodes, called the *origin* and *destination*. Associated with each edge is a *distance*, a nonnegative number. The objective is to find a shortest path between the origin and destination.

There is a very fast algorithm for this problem. The essence is to explore outward from the origin, successively finding the shortest paths to nodes in the network until the destination is reached.

Consider the park example, with origin **O** and destination **T**. Looking just locally, we can determine that we can reach the following nodes at the given distances:

- **A**: distance 2
- **B**: distance 5
- **C**: distance 4

Now consider the closest node, in this case node **A**. We can reach it directly in distance 2. Might we find another path from **O** to **A** that is shorter? Of course not! We would have to go through another node, and we already know that all other nodes are at least distance 2 away from **O**. Therefore we know that **A** is the closest node to **O** and has distance 2.

What is the second closest node to **O**? The key to finding this is to know the following (due to Dijkstra):

*If we have found the **k**th closest nodes to **O**, then the **k+1**st closest has a path that goes from **O** to one of the **k** closest and then a single edge from that node to the **k+1**st.*

This assumes we treat node **O** as the 0th-closest node to **O**. In other words, to determine the second closest, look at edges going out from **O** and the first closest:

- **B** has distance 4 (2 to get from **O** to **A** and 2 to get from **A** to **B**)
- **C** has distance 4 (direct from **O**)
- **D** has distance 9 (2 to get from **O** to **A** and 7 to get from **A** to **D**)

and those are all the possibilities. The shortest of these is the second closest. There is a tie, so we will arbitrarily make **B** the second closest (distance 4) and **C** the third closest (distance 4).

Which node is the fourth closest? The possibilities are:

- **E** at distance 7 (through **B**)
- **D** at distance 8 (through **B**)

so the fourth closest is **E**. The fifth closest is either

- **D** at distance 8 (through **B** or **E**), or
- **T** at distance 14 (through **E**).

The fifth closest is **D**. So the sixth closest is **T** at distance 13 through **D**. Working our way backwards gives us that the shortest path from **O** to **T** is

> O-A-B-D-T (or, alternatively, O-A-B-E-D-T).

In general the algorithm is as follows:

> *To find the **k+1**st closest given the **k** closest:* Call the **k** closest (including the origin) the ''solved'' nodes and the others the unsolved nodes. For each solved node, find the unsolved node connected to it by the shortest edge. This is a candidate node. For each candidate, add the distance on the edge to the distance from the origin to the solved node. The candidate with the shortest such distance is the **k+1**st closest node.

You can find the shortest path from the origin to the destination by repeatedly using the above step until the destination becomes a solved node. Ties may be broken arbitrarily, and may provide information about alternative shortest paths.

Discussion. It is important to recognize that the shortest path problem is a *model* that can be used even if no distances are involved. For instance, the numbers on the edges might represent costs, so the problem is finding a sequence of tasks that completes some goal at minimum cost. Or the numbers might be time, and the problem is minimizing the amount of time required to accomplish some goal. This is illustrated by the following exercise:

Exercise 1 *At a small but growing airport, the local airline company is purchasing a new tractor for a tractor-trailer train to bring luggage to and from the airplanes. A new mechanized luggage system will be installed in 3 years, so the tractor will not be needed after that. However, because it will receive heavy use, and maintenance costs are high, it may still be more economical to replace the tractor after 1 or 2 years. The following table gives the total net discounted cost associated with purchasing a tractor in year i and trading it in in year j (where year 0 is now):*

	j = 1	2	3
i = 0	8	18	31
1	—	10	21
2	—	—	12

The problem is to determine at what times the tractor should be replaced (if ever) to minimize the total costs for tractors.

(a) Formulate this problem as a shortest path problem.

(b) Use the algorithm presented to solve this problem.

Exercise 2 *Solve the shortest path problem for the network in figure 2.*

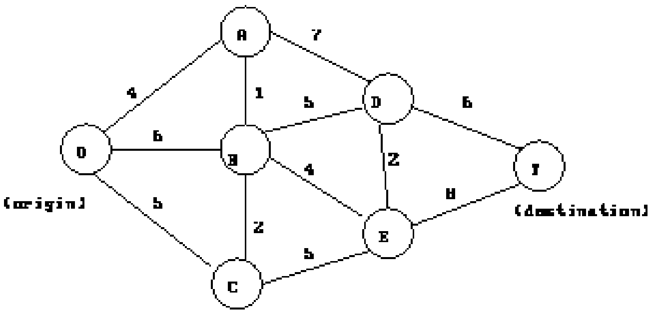

Figure 2: Distances

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