Bellman Equation

Matthew Cocci

November 22, 2013

1 Introduction

The shortest path problem has the following features:

1. You want to get from point 1 to N.
2. There are a number of possible nodes in between 1 and N, permitting a number of different possible paths you might take.
3. The paths from node to node have different costs.

Therefore, we want a general solution to the problem that provides us with the *least cost* path to travel from 1 to N.

2 The Bellman Equation

Suppose that at every single node, $i \in \{1, \ldots, N\}$, we *knew* the least-cost path to get from node i to node N. Of course we don't, unless we're trivially already at node N, but just suspend disbelief for a second and suppose we do.

Now since we know the least-cost paths, we know the "best-case" cost to get from node i to N, for all i. Denote that "best-case" cost by $J(i)$. Now it makes sense that this function J, for each i, should also satisfy what's called the *Bellman Equation*:

$$J(i) = \min_{j \in F_i} \{c(i, j) + J(j)\} \tag{1}$$

In words, this equation says

1. To find the best-case cost to go from i to N, consider the set of nodes that you can reach from node i, denoted F_i.
2. Next, consder the cost of jumping from node i to node j, denoted $c(i, j)$.
3. Finally, since you know the best-case cost, $J(j)$ for $j \in F_i$, pick the minimum of $c(i, j)$ plus $J(j)$.

The best solution to our problem must, therefore, satisfy Equation 1.

3 Solving for J

We now detail the standard algorithm to find J.

1. For some large M, set
$$J_0(i) = \begin{cases} M & i \neq N \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

2. Next, set $J_{n+1}(i) = \min_{j \in F_i} \{c(i,j) + J_n(j)\}$ for all i.

3. If $J_{n+1} \neq J_n$, increment n and return to step 2.

The solution thus propagates back from the destination node, N.