# THE BELLMAN-FORD ALGORITHM : AN INTRODUCTION

The Bellman-Ford algorithm (also called the Ford-Fulkerson algorithm) is based on a principle that is intuitively easy to understand: If a node is in the shortest path between A and B,

then the path from the node to A must be the shortest path and the path from the node to B must also be the shortest path. As an example, suppose that we want to find the shortest path from node 2 to node 6 (the destination) in Figure 1.28. To reach the destination, a packet from node 2 must first go through node 1, node 4, or node 5. Suppose that someone tells us that the shortest paths from nodes 1, 4, and 5 to the destination (node 6) are 3, 3, and 2, respectively. If the packet first goes through node 1, the total distance (also called total cost) is 3 + 3, which is equal to 6. Through node 4, the total distance is 1 + 3, equal to 4. Through node 5, the total distance is 4 + 2, equal to 6. Thus the shortest path from node 2 to the destination node is achieved if the packet first goes through node 4.

To formalize this idea, let us first fix the destination node. Define $D_j$ to be the current estimate of the minimum cost (or minimum distance) from node j to the destination node and $C_{ij}$ to be the link cost from node i to node j. For example, defined to be zero (that is, $C_{ii}$ f1 0), and the link cost between node i and node k is infinite if node i and node k are not directly connected. For example, $C_{15}$ f1 $C_{23}$ f11 in Figure 1.28. With all these definitions, the minimum cost from node 2 to the destination node (node 6) can be calculated by

$$D_2 \text{ fl } \min\{C_{21} + D_{1,}\ C_{24} + D_{4,}\ C_{25} + D_5\}$$

$$\text{fl } \min\{3 + 3, 1 + 3, 4 + 2\}$$

$$\text{fl } 4$$

Thus the minimum cost from node 2 to node 6 is equal to 4, and the next node to visit is node 4. One problem in our calculation of the minimum cost from node 2 to node 6 is that we have assumed that the minimum costs from nodes 1, 4, and 5 to the destination were known. In general, these nodes would not know their mini-mum

costs to the destination without performing similar calculations. So let us apply the same principle to obtain the minimum costs for the other nodes. For example,

$$D_1 \text{ fl } \min\{C_{12} + D_2, C_{13} + D_3, C_{14} + D_4\}$$

And $$D_4 \text{ fl } \min\{C_{41} + D_1, C_{42} + D_2, C_{43} + D_3, C_{45} + D_5\}$$

A discerning reader will note immediately that these equations are circular, since $D_2$ depends on $D_1$ and $D_1$ depends on $D_2$. The magic is that if we keep iterating and updating these equations, the algorithm will eventually converge to the correct result. To see this outcome, assume that initially $D_1$ fl$D_2$ fl . . . fl $D_5$ fl $\infty$. Observe that at each iteration, $D_1, D_2, \ldots, D_5$ are nonincreasing. Because the minimum distances are bounded below, eventually $D_1, D_2, \ldots, D_5$ must converge.

Now if we define the destination node, we can summarize the Bellman-Ford algorithm as follows:

1. Initialization

$$D_i \text{ fl} \infty; \text{ for all } i \neq d$$

$$D_d \text{ fl } 0$$

2. Updating: For each $i \neq d$,

$$D_i \text{ fl } \text{mi}_j\text{n}\{C_{ij} + D_{j\}},\ \text{for all } j \neq i$$

Repeat step 2 until no more changes occur in the iteration.