Course: Introduction to Graph Algorithms

Professor: C Pandu Rangan

Department: Computer Science and Engineering

Institute: IISc

Week: 03

Lecture 12 Belman Equation Part 3

So if the graph has no negative cycle or 0 cycle, we know that the solution of this equation called Bellman equation is unique, xv equal to minimum over all uv belonging to E, x of u plus weight of uv.

$$x_s = 0$$

$$x_v = \min_{(u,v) \in E} \{ x_u + w(u,v) \}$$

It is a very interesting equation, I know that it is a solution is what I want, but how do I go about solving this equation? This equation has got circular dependencies, we have already seen a small example, in fact if the original graph has a kind of a cycle, the variables which are depending on in that chain will display a cyclic dependency. If there is a cyclic dependency things get messy and then when you have large number of equation you do not have any systematic procedure to solve this set of equations. So after doing all this nice work, finally we have in our hand a system of equations which are difficult to solve or there is no known procedure available to solve them. So how do we circumvent this situation? So mathematicians have come up with a brilliant idea, okay. The method that they approach they take whenever you see this kind of a non-linear and complicated equations, they take an approach called iterative approach, this also called as method of iterated improvements or method of successive approximations.

These are all they all mean the same thing okay. So what we do in an iterative approach is the following. We start with some tentative solution and we find a better solution. So we will have a method that is going to improve in some sense the solution in our hand.

So we start with a solution, apply that method, get a better solution. Again apply the same method on the better one you got and you get much better one. So in this way you keep improving the solutions. So there is a basic method which will define the so called the improvement step and that basic procedure is going to be repeatedly applied. So you get a solution s0, from that you get a solution s1 and from that you get solution s2 and so on okay, each is an improvement over the previous one.

$$s_0 \rightarrow s_1 \rightarrow s_2 \cdots$$

So I am not directly solving it, I am defining a procedure that will slowly take me to the solution, okay? So I am taking a roundabout route, I do not have a direct computational procedure to solve the equation, but what I do is an iterative method, okay, a very clever approach. In numerical methods and in various other branches of mathematics we might have seen these kind of techniques. What is important for us is that, the this process of improvement should quickly converge, in fact ideally it should converge in finite number of steps and that finite number should also be small so that we quickly reach the solution. The hope is that, when you keep on improving it you get to reach the solution, right, this should converge to a solution all these things should happen. In our context we are looking for the shortest path.

So what we can do is the following, we should start with some path do some computation and try to obtain a better path, and again do the same kind of a computation on that path and obtain a still better path or a shorter path. So, in this way we may plan to obtain a series of paths, one shorter than the previous one, with the hope that this process leads to the shortest path, is called the method of iterated improvement brilliant idea. But then identifying in what way we can achieve the improvement, and how many iterations we require, all these things are technical details and we are going to take a look at them now, okay?

So one natural way to go towards the solution is through what is known as bounded paths. We will find it somewhat difficult to directly deal with paths. So, we will go through what is known as walks, bounded walks, but bounded walks and bounded paths are related and we use all of that and finally solve the problem, okay? It is a very high level description we are going to see the details now, let Wkv denote the set of all walks from s to v containing utmost k edges, here we have k greater than or equal to 0, okay?

$$W_k(v)\ k \text{ edges}, \quad (k \geq 0)$$

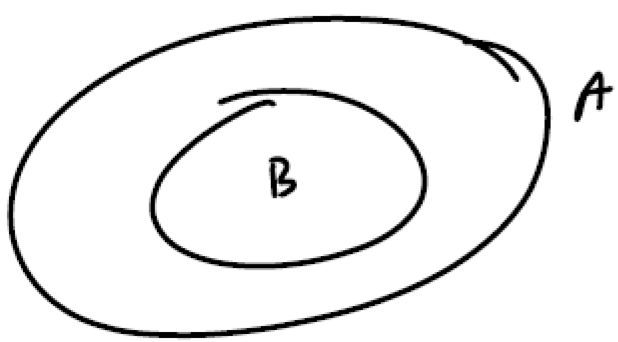
Set of all walks, you can walk from s to v, a series of edges end at v. That series can have repetition and it is a walk after all okay. So Wkv is the set of but k is a finite number therefore, this set is a finite set, Wkv is a finite set, it could be empty but it is a definitely finite, because the number of edges you can have is bounded, that is why these are called bounded walk. You cannot have infinitely many edges here, you can have only you are allowed to have only at most k edges. Because it is a finite set, each walk has a weight, there is a walk with minimum weight right. Let alpha kv denote the weight of shortest walk in Wkv, okay,

$$\alpha_k(v)$$

this is the weight of the shortest walk in Wkv, it is a number, in our case it is an integer right. What are the properties we can observe about this? First of all, notice that Wkv is a subset of Wk plus 1V okay,

$$W_k(v) \subseteq W_{k+1}(v)$$

why? Every path that has got utmost k edges obviously will be in the set of walks that has got utmost k plus 1 edges. A walk that has got less than or equal to k edges is automatically a walk that has got less than or equal to k plus 1 edges okay, a walk with utmost k edges is automatically a walk with utmost k plus 1 edges. So every walk here, every walk in Wkv will be a walk in Wk plus 1v, every walk here, and Wk plus 1v may have some more walks because you may take a slightly a round about and then reach v and thereby, with k plus 1 edges there may be a walk, in which case this will be a superset. Otherwise it will be equal or it is a superset. So you have a set B and you have a superset A.



What can you say about the minimum of these 2? Suppose B is a set of numbers, A is a super set of numbers, okay, just numbers, B is a set of numbers and so minimum of A will be less than or equal to minimum of B. Okay,

$$\text{Min } A \leq \text{Min } B$$

you can put in this way also, minimum of B is greater than or equal to minimum.

$$\text{Min } B \geq \text{Min } A$$

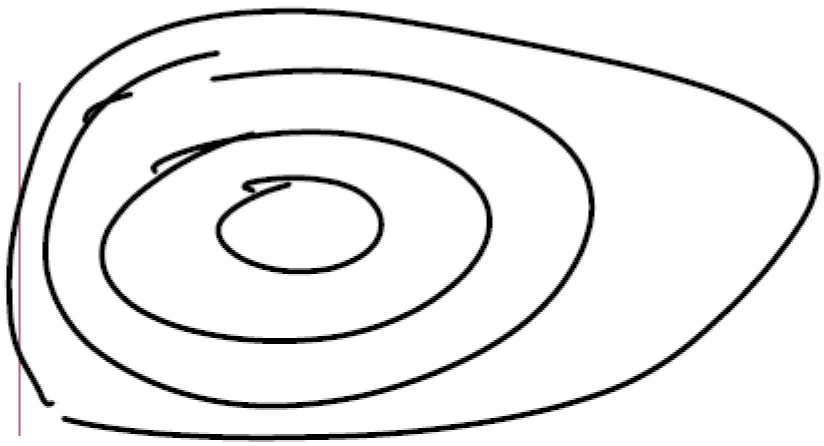
That is because every element of B is an element of A, therefore the smallest element of B is also there in A. Minimum of A is smaller than everybody, so it will be smaller than even minimum of B, that is all the simple logic, very simple logic. A superset of numbers probably will have a smaller minimum, it will have the same minimum or it may have a smaller minimum okay. Because of this, you immediately see that alpha k v is greater than or equal to alpha k plus 1 v.

$$\alpha_k(v) \geq \alpha_{k+1}(v)$$

It is a minimum over a bigger set, a minimum over a bigger set may have a smaller minimum. That is the reason why this will be true, that means we have a decreasing sequence of values okay. alpha 0v is greater than or equal to alpha 1v which is greater than or equal to alpha 2v and so on.

$$\alpha_0(v) \geq \alpha_1(v) \geq \alpha_2(v) \geq \cdots$$

You have a decreasing sequence of values defined by this collection of sets. This is a hierarchy of set, it is a collection of set one containing other, okay, Wk so I have this set and then something larger and something and something bigger.



So in this way, it keeps containing the previous ones. Therefore, the minimums will form a decreasing sequence. We are going to define for paths similar quantities. Okay what are the let P kv is set of all P is a path from s to v with utmost k edges, of course K is greater than or equal to 0.

$$P_k(v) = \{ P \mid \quad k \geqslant 0$$

So here also we have the following, Pkv is a subset of Pk plus 1v.

$$P_k(v) \subseteq P_{k+1}(v)$$

Because a path with less than or equal to k edges, is automatically it has got less than or equal to k plus 1 edges, so that path will be in the other set also, so every element of Pkv is an element of Pk plus 1v, Pk plus 1v may have additional paths, right. So, it is a hierarchy of set, one containing the previous one. Now we are going to define delta kv equal to the weight of shortest path in Pkv.

$$\delta_k(v) = \text{weight of shortest path in } P_k(v)$$

Pkv is a finite set, each path in Pkv has got a weight, therefore there is a shortest path minimum weight. That minimum weight is delta kv, that minimum weight is actually delta kv. The alpha sequence can be an infinite sequence, but the delta sequence is a finite sequence. Delta 0v is greater than or equal to delta 1v, greater than or equal to delta n minus 1 v and there it stops, after that everything will be equal, delta nv is equal to delta n plus 1v and so on.

$$\delta_0(v) \geqslant \delta_1(v) \geqslant \cdots \geqslant \delta_{n-1}(v) = \delta_n(v) = \delta_{n+1}(v) \ldots$$

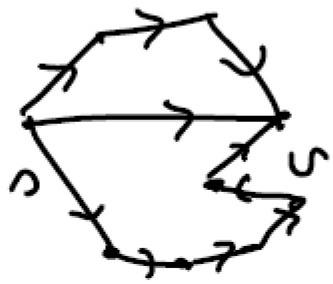
It is not an infinite sequence, it stops at n minus 1, you can define but all of them will be equal to delta n minus 1 only, it would not decrease any further why? It is a path, any path, even the longest path cannot have more than n minus 1 edges. In a path, no vertex and no edge can be repeated, therefore the maximum number of edges in any path in a graph with vertex set of size n, suppose vertex set has got n vertices, what is the longest path? n minus 1, any path you take from anywhere to anywhere, it cannot have more than n minus 1 edges, okay, the maximum number of any path is n minus 1.

$$|V| = n, \text{ is } (n-1)$$

That means every path is going to have less than or equal to n minus 1 edges, what does this mean? Every path will have only less than or equal to n minus 1 edges. So P n minus 1 v is set of all paths. This is set of all paths from s to v, this is set of all paths from s to v. The minimum weight in that is, the shortest path from s to v. Therefore delta n minus 1v is in fact delta v.

$$\delta_{n-1}(v) = \delta(v)$$

You are finding the minimum over all paths and that is the shortest path, okay. Therefore, delta v is delta n minus 1v. What can you say about delta nv? Set of all paths with less than or equal to n edges. This is same as because this itself contains all paths. It cannot have any superset. So all higher numbered indices are going to define the same set of paths, okay, because Pn minus 1 v itself captures all paths, okay? So for example from s to v, this is a path of length 1, this is a path of length 3, okay, if there is a path like this 1, 2, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6 this is a path of length 6, so you have various paths of various lengths, number of edges.



But none of them can have more than n minus 1, that is the reason why you have delta 0v greater than or equal to delta 1v, this can so this a path, you have a shortest path among p1, shortest path among p2, shortest among p2, but shortest among p n minus 1 is the shortest path. delta n minus 1v which is equal to delta v

$$\delta_0(v) \geq \delta_1(v) \geq \cdots \delta_{n-1}(v) = \delta(v)$$

and this is equal to delta nv, this is equal to delta n plus 1v, and so on.

So this sequence becomes stationary here. So you are going to, when you are going to consider the shortest paths, in these sets their lengths, become shorter and shorter, initially you have this, then you have a better path and you have. So find the shortest path in p1, using this find a shortest path in p2, using that find a shortest path in p3.

$$\begin{aligned} \delta_0(v) \geq \delta_1(v) \geq \cdots \delta_{n-1}(v) &= \delta(v) \\ &= \delta_n(v) \\ &= \delta_{n+1}(v) \\ &\;\;\vdots \end{aligned}$$

In this way when you go, you get shorter and shorter path after n-1 stages, you have the shortest path in your hand, method of iterated improvement, okay? Therefore, we have a high level plan where we are going to keep finding the shortest paths in this collection, which keeps giving us better and better paths. There is a minor technicality here, we will have some difficulty in working with the collection of paths, but we can work comfortably in the collection of walks. We can work in the collection of walks easily. But working in the collection of paths is a bit tricky. We will see why we have introduced the alphas and deltas. Our goal is to find delta v, which means our goal is to find delta n minus 1. but we are going to find a delta n minus 1 not via delta 0, delta 1, delta 2 but via alpha 0, alpha 1 and so on and then relate these two and get to our solution, okay? We will see the details of these in our next session, thank you.