Course: Introduction to Graph Algorithms

Professor: C Pandu Rangan

Department: Computer Science and Engineering

Institute: IISc

Week: 03

Lecture 13 Belman Equation Part 4

So this is a decreasing sequence of numbers and this can go up to minus infinity, however whatever is the graph whether it has a negative cycle or positive cycle it really does not matter what kind of graph it is. The delta sequence is always finite, delta 0v is greater than or equal to delta 1v is greater than or equal to delta n minus 1v and it stops here and after that all values are equal.

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

$$\alpha_0(v) \geq \cdots \quad \cdots \geq \cdots \quad -\infty$$

That is because as we have seen already. the maximum number of edges in any path is just n minus 1. That is the reason why this will not extend any further. You will have only equal values, it will not descend beyond this point. So we write like this, equal to delta nv, equal to delta n plus 1v, and so on for all of them it is equal.

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

How are alpha values and delta values are related? Our goal is the delta values, so we will see how alphas and deltas are related. Since every path is a walk, since every path is a walk, it is clear that Pk(v) is a subset of Wk(v). A path that has got at most k edges is also a walk that has got at most k edges. Therefore, every element of Pk is an element of Wk, Wk may have more elements. So, because of this, the bigger sets will have smaller minimum value right. alpha k(v) is less than or equal to delta k(v), alright.

$$P_k(v) \subseteq W_k(v)$$
$$\alpha_k(v) \leq \delta_k(v).$$

So alpha and delta are related this way. There is another interesting fact, is always greater, if G has no negative cycle then alpha k(v) is equal to delta k(v).

$$\alpha_k(v) = \delta_k(v)$$

That is because, if a graph has no negative cycle, every shortest walk will indeed be a shortest path, if G has no negative cycles, every, and also we will well we will avoid 0 cycle in the sense 0 cycle it just can be simply removed without changing the values okay, every shortest walk is indeed a shortest path, okay. This can be easily proved by contradiction. If the shortest walk has a repetition or as a cycle that can be removed and the weight can be further reduced. Let us say s to v is a cycle, is a walk, I can remove this cycle of edges and still have a walk from s to v.

And that walk will have a smaller weight. That is because I am removing a cycle. I am assuming that G has no 0 cycle, no negative cycle, so cycle if there is any they are all positive, so I am reducing by a positive quantity, and that will be a contradiction to the minimality. So, if the shortest walk has a cycle, it can be reduced and then you can find a shorter walk and that is not possible. That is the reason why shortest walk cannot contain any cycles. That means it is indeed a path. No repetition of vertices or edges therefore, that is the reason why we have this equality, okay. This is very interesting alpha k(v) equal to delta k(v). Now we will take a closer look at the definition of alpha k for k equal to 0 and v equal to s and those edge cases; are to be sorted out. Okay, so, what can we say about alpha k(s)?

$$\alpha_k(s) \quad ?$$

What do you mean by that? s to s there is a walk, that means it is a closed walk, okay, this is a closed walk. And what can we say about delta k(s) this is a cycle close to path, this is, okay.

$$\delta_k(s) - \text{cycle}$$

Now the case when k is 0, what can we say about delta 0(s)?

$$\delta_0(s) \ ?$$

How do we define this less than or equal to 0 edges no edges from S to S okay without any edges it is possible for you to go just do not move out of S right and therefore since there is a possibility except that you are not using any edge or anything okay. So the weight is assumed to be 0 delta 0s is 0 same for alpha 0s is set to 0.

$$\delta_0(s) = 0$$
$$\alpha_0(s) = 0$$

What can we say about alpha kv when k equal to 0 that is alpha 0V.

$$\alpha_k(v), \quad k = 0 \ ?$$

So from s to v you have to go with 0 edges it is not possible. a walk does not exist with 0 edges right there is no way with 0 edges you can go from s to v where v is not equal to s right. So alpha 0 v equal to infinity since a walk from s to v.

$$\alpha_0(v) = \infty$$

v not equal to s it is some other how will you go with 0 edges there is no way therefore such a path does not exist since a walk from with 0 edges does not exist, whenever something does not exist path or walk we said to be similarly delta 0v equal to infinity.

So this remember these boundary cases. So in some alpha 0 s equal to 0. alpha 0 v equal to infinity for v not equal to s.

$$\alpha_0(s) = 0$$
$$\alpha_0(v) = \infty, \quad v \neq s$$

Similarly delta 0 s equal to 0 delta 0 v equal to infinity for v not equal to s.

$$\delta_0(s) = 0$$
$$\delta_0(v) = \infty \quad v \neq s$$

If G has no negative cycles then alpha 0 v equal to delta 0 v this will be greater than or equal to alpha 1 v but alpha 1 will be equal to delta 1 v and this will be greater than or equal to alpha n minus 1v which is equal to delta n minus 1v for every vertex

$$\begin{array}{ccccc}
\alpha_0(v) & \geq & \alpha_1(v) & & \geq \alpha_{n-1}(v) \\
\parallel & & \parallel & \cdots & \parallel \\
\delta_0(v) & & \delta_1(v) & & \delta_{n-1}(v)
\end{array}$$

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

And we stop with this there is no need to proceed further if you consider those further elements they will all be equal to these elements. and delta n minus 1v we know is a delta v therefore alpha n minus 1v equal to delta n minus 1v equal to delta v.

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

This is the important observation and we will see how we are going to use this. since alpha n minus 1 equal to delta v and notice that alpha is a decreasing sequence therefore I have a decreasing sequence leading to my answer. In other words something can be made smaller and smaller in n minus 1 stages I get the answer. So if I have a method of improvement which is using alpha i to compute alpha i plus 1, I can apply that procedure again and again I will start with alpha 0 compute alpha 1 I apply the same procedure start from alpha 1 and obtain alpha 2 apply the same procedure again on alpha 2. and get alpha 3 in this way I can keep on computing the values one smaller than the other or one solution better than the other and in n minus 1 stages I am going to hit the answer that I am looking for. So we have to look into the technical details of how alpha i plus 1 and alpha i are related. If I could explicitly derive the relation then I can device a computational process which will use that relation and obtain the values okay. Fortunately we have the alpha values satisfying the equations much like the Bellman equation okay.

Let us recall what the Bellman equations are, the Bellman equations are delta v is equal delta v is less than or equal to delta u plus weight of uv and there is an edge such that delta v equal to delta u plus weight of uv this is for all uv there exist uv satisfying this condition. together we got delta v equal to minimum of delta u plus weight of uv.

$$\left.\begin{array}{l} \delta(v) \leq \delta(u) + w(u,v) \quad \forall (u,v) \\ \delta(v) = \delta(u) + w(u,v) \quad \exists (u,v) \end{array}\right.$$

$$\delta(v) = \text{Min}\{\delta(u) + w(u,v)\}$$

We are going to get a similar relation so these are called Bellman equations for bounded walks for minimum weight bounded walks. Here I am not considering all walks this is a minimum across all paths delta value. I am also I am now focusing on alpha but again alpha with respect to a k we will show that, theorem alpha i plus 1v is less than or equal to alpha i v less than or equal to alpha i u plus weight of uv for all edge uv this will be true. okay this is true for all edge uv in E

$$\alpha_{i+1}(v) \leq \alpha_i(u) + w(u,v)$$
$$\forall (u,v) \in E.$$

but there is an edge there exist an edge uv such that alpha i plus 1v is exactly equal to alpha iu plus weight of uv.

$$\exists (u,v) \ni$$
$$\alpha_{i+1}(v) = \alpha_i(u) + w(u,v)$$

These two together imply that alpha i plus 1 v is equal to minimum over all edges uv such that delta u plus weight of uv.

$$\Rightarrow \alpha_{i+1}(v) = \min_{(u,v)\in E} \{\alpha_i(u) + w(u,v)\}$$

Consider all the incoming edges to v each incoming edge to v is going to define one value take the minimum over all you are getting this identical to the Bellman equation for weights of the shortest paths. These are the weights of the shortest walk among the bounded walks, walks with bounded length. While the Bellman equation for shortest path weights they have circular dependency nonlinearity and so on there is nothing like that here. Left hand side has got n values to be computed but they are depending on a completely different set of n values okay. So if I have oh I am sorry this is not delta I this is alpha i. Right hand side has got alpha i values, left hand side has got alpha i plus 1 values. So I have a vector of values alpha i s alpha i of other vertices okay

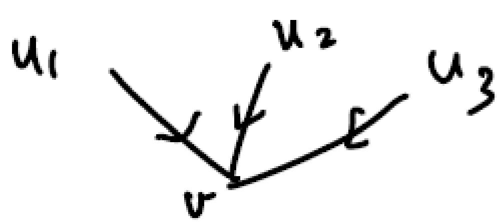
$$\alpha_i[s],\ \alpha_i[\ ],\ \ldots$$

there is a vector of values using this I can obtain the another vector of values alpha i plus 1 s alpha i plus, so on

$$\alpha_{i+1}[s],\ \alpha_{i+1}[\ ],\ \ldots$$

So if I have one array of values I can compute another array of values without any clash cyclic dependence and everything there is a formula. What are all the values in this array I have to consult to construct one element of the next array. In order to compute alpha i plus 1 of a particular vertex okay what are all the alpha i values I would need okay.

For example if I have u1, u2, u3 these are all the incoming edges to v



then in order to compute alpha i plus 1 v I need alpha is alpha iu1 I need alpha i u2 I need alpha iu3 I need similarly alpha iu4 okay u3 is enough.

$$\alpha_i[s] \cdots \alpha_i[u_1] \cdots \alpha_i[u_2] \cdots \alpha_i[u_3] \cdots$$

$$- \quad -- \qquad \underline{\underline{\alpha_{i+1}[v]}}$$

So in order to compute this value I need this, this one and this one which are all available okay. So I am assuming that alpha i array of values available then the array is indexed by vertices the array is indexed by vertices for example if the vertex at V is v1, v2 etcetera vn

$$V = \{ \underset{\substack{\downarrow \\ s}}{v_1}, v_2, \ldots v_n \}$$

then I have alpha i s let us say v1 is s then alpha i v1 alpha i v2 then I will have alpha i vn

$$\alpha_i[v_1], \alpha_i(v_2], \ldots \alpha_i[v_n]$$

all these values I assume I have then I can compute alpha i plus 1 v1 alpha i plus 1 v2 alpha i plus 1 vn these n values I can compute using these n values okay.

$$\alpha_{i+1}[v_1], \quad \alpha_{i+1}[v_2], \ldots \quad \alpha_{i+1}[v_n]$$

How do you compute? Here is the formula okay, alpha i plus 1v is minimum of alpha iu plus wuv, weight of the edge is there

$$\Rightarrow \alpha_{i+1}(v) = \min_{(u,v) \in E} \{\alpha_i(u) + w(u,v)\}$$

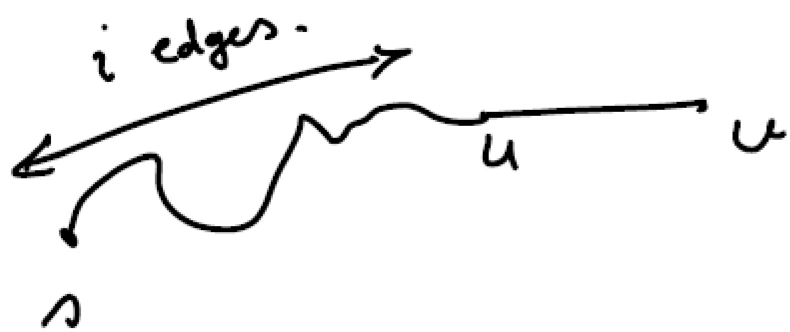
and alpha i values are there in one array. Use the values in that array to build this array and all alpha i values are smaller than all alpha i plus 1 values are smaller than alpha i values. Therefore, this is an improvement compute the array of values alpha i plus 1 by using array of values alpha i this array and this is called improvement step or improvement procedure. I have a computation, set of computation I will do and if you look into the definition this theorem it is saying that alpha i plus 1 is minimum among these values okay.

$$\alpha_{i+1}(v) \leq \alpha_i(u) + w(u,v)$$
$$\forall\ (u,v) \in E.$$

We have already seen this being a subset alpha i plus 1v could be alpha iv or minimum of this right. the shortest walk in W i plus 1 v has less than or equal to i edges it will be a shortest walk even in Wiv hence in this case alpha i plus 1v is equal to alpha iv

$$\alpha_{i+1}[v] = \alpha_i[v]$$

if it has got exactly i plus 1 edges because it can have up to. i plus 1 edges. Now assume that the shortest walk has exactly i plus 1 edges. So I have a shortest walk, this has got s to v is the last edge in the walk this s to v walk has got i plus 1 edges therefore this part. will have i edges this part will have i edges



we will use a simple notation let Wsv be a shortest walk from s to v with i plus 1 edges

$$W[s,v]$$

then Wsv can be written as Wsu plus uv. Wsu plus uv the portion of the walk from s to u and the edge uv.

$$W[s,v] = W[s,u] + (u,v)$$

We claim this is a shortest walk. We claim that this is a shortest walk. If this is not the shortest walk. this called cut and paste argument okay. The style the proof style is known as cut and paste argument if Wsu is not a shortest walk. W dash su be a shortest walk okay.

$$W'[s,u]$$

I will draw W dash also in the next s to uv this is Wsu, W dash might be something like this. This is W dash, this is W.

Weight of W dash su is less than weight of W su right.

$$w(W'(s,u)) < w(W(s,u))$$

That is because this is not the shortest W is not the shortest W dash is the shortest. W dash is the shortest walk with W dash as less than or equal to i edges.

$$|W'| \leq i \text{ edges}$$

Now I can cut and paste. I can remove Wsu include W dash so from su go through W dash and go to v so weight of W dash plus uv is equal to weight of W dash plus weight of uv but this is strictly less than weight of W plus weight of uv okay sorry weight of W dash ends at u. s u plus weight of uv. which is equal to weight of Wsv which is equal to alpha this is a shortest path right so it is length it is weight is going to be alpha i plus 1v

$$\begin{aligned} w(W' + (u,v)) &= w(W') + w(u,v) \\ &< w(W[s,u]) + w(u,v) \\ &= w(W(s,v)) \\ &= \alpha_{i+1}[v] \end{aligned}$$

this is a contradiction I have another walk with a smaller weight. okay it is a length is also less than or equal to i plus 1 and it is weight is strictly smaller it is a contradiction this contradicts the minimality of W okay we have started with the shortest walk W is the shortest walk and W dash plus uv is a shorter walk not possible shorter than the shortest is not possible this contradicts the minimality of W okay. W dash has got less than or equal to i edges, W dash plus uv is less than or equal to i plus 1 edges. Here is a walk with less than or equal to i plus 1 edges and whose weight is smaller than alpha i plus 1v not possible therefore such a W dash cannot exist this implies such a W dash cannot exists implying Wsu is a shortest walk and it has less than or equal to i edges this implies weight of Wsu equal to alpha iu. and this implies weight of Wsu which is equal to alpha i plus 1 v. which is equal to weight of Wsu plus uv which is equal to Wsu we have shown is alpha iu plus weight of uv.

$\Rightarrow$ such a W' can not exist

$\Rightarrow$ $W(s,u)$

$\leq i$ edges

$$\Rightarrow \; w(W(s,u)) = \alpha_i(u)$$

$$\Rightarrow \; w(W(s,u)) = \alpha_{i+1}(v)$$

$$= w(W(s,u) + (u,v))$$

$$= \alpha_i(u) + w(u,v)$$

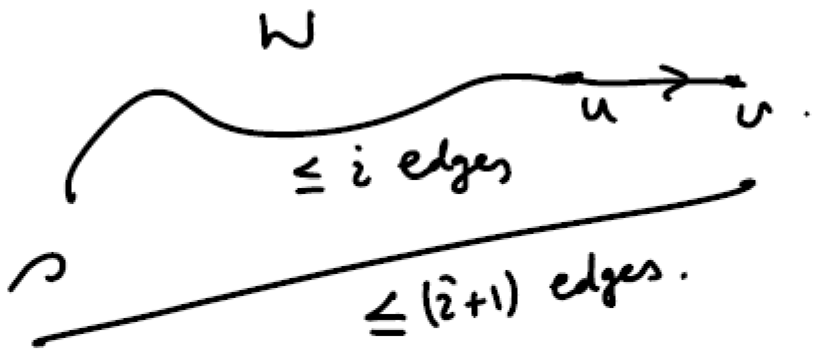
we have shown that alpha i plus 1v is equal to alpha iu plus weight of uv for some edge uv.

$$\alpha_{i+1}(v) = \alpha_i(u) + w(u,v)$$

It is very easy to show that alpha i plus 1v is less than or equal to alpha iu plus weight of uv

$$\alpha_{i+1}(v) \leq \alpha_i(u) + w(u,v)$$

that is because you take any w s to u this is w and it has got less than or equal to i edges extended by v at the edge uv. This is a walk the total thing is a walk with less than or equal to i plus 1 edges.



Therefore the weight of that walk must be larger than or equal to alpha i plus 1v. What is alpha i plus 1v that is the smallest weight this is some walk. therefore its weight will be larger that means alpha i plus 1 v is less than or equal to w has got alpha iu the edge you have added is weight of uv and this proves the second part. okay

$$\alpha_{i+1}(v) \leq \alpha_i(u) + w(u,v)$$

So take any shortest walk and you are going to get a walk with less than or equal to i plus 1 edges that is a arbitrary walk the weight of an arbitrary walk must be larger than or equal to weight of the minimum or the shortest walk therefore this follows okay hence we have proved that alpha i plus 1 v is equal to minimum over all uv such that alpha iu plus weight of uv.

$$\alpha_{i+1}(v) = \min_{(u,v)\in E} \left\{ \alpha_i(u) + w(u,v) \right\}$$

Therefore no circular dependency and once I have an alpha i array of values I can compute alpha i plus 1 array of values. I know alpha0 array of values alpha0 s equal to 0 alpha0 v equal to infinity

$$\alpha_0(s) = 0$$

$$\alpha_0(v) = \infty$$

Therefore we know alpha0 array of values using this obtain alpha1 array of values. using this you obtain so you go through n minus 1 stages okay you go through n minus 1 stages alpha array to alpha 1 array to alpha n minus 1 array. So stop after n minus 1 improvement steps.

$$\begin{array}{c} \alpha_0 \\ \downarrow \\ \alpha_1 \\ \vdots \\ \downarrow \\ \alpha_{n-1} \end{array}$$

You would have got alpha n minus 1v we know that since the graph has no negative cycle this is in fact delta n minus 1v we also know that delta n minus 1v is in fact delta v therefore you can output the alpha n minus 1v values. very simple algorithm, iterative procedure.

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

Getting this delta v from Bellman equation was hard getting this values using the Bellman like equations on bounded walk lengths, walk weights we are able to get an iterative algorithm, it is improving the weights of the walk and finally it converges to the answer we are looking for. Though we are working in the class of walks the shortest walk being shortest path. We are able to show that the convergence of that sequence is same as the convergence of the sequence that we want note that this is not true for arbitrary graph, for arbitrary graph if you apply this kind of a improvement step you are not going to end up with shortest path weights, you will get some values but they will not be shortest path weights. However, if G has no negative cycle or 0 cycle applying n minus1 improvement steps. lead to the answer we are looking for this is the key idea behind Bellman and Ford algorithm. We will take a detailed look at Bellman and Ford algorithm in our next session thank you.