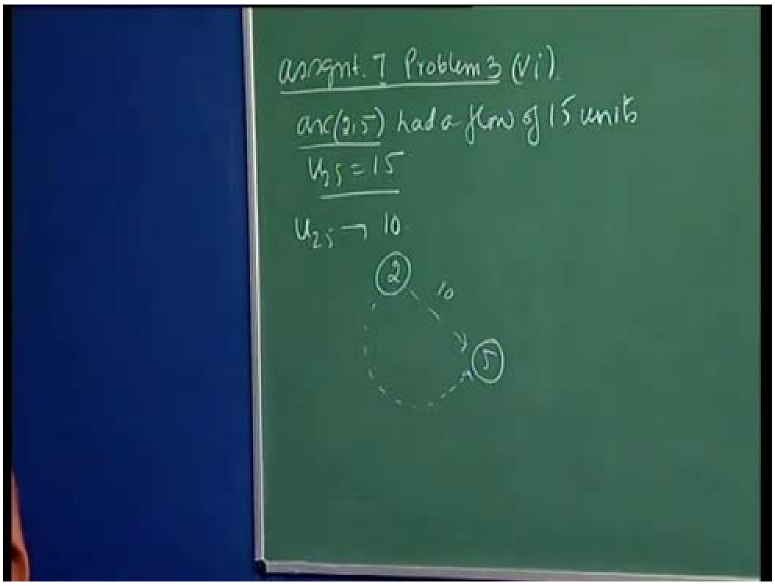
Linear Programming and its Extensions.
Prof. Prabha Sharma
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Module No. # 01
Lecture No. # 34
Problem Three (assignment seven) Min-Cut Max-Flow Theorem Labelling Algorithm.

Let me revise it assignment 7 just for a moment, I discussed it in the last lecture, and in problem 3 part 6, I had asked you to consider the case when the capacity is reduced, so arc 2, 5 had a flow of 15 units in the optimal solution and u 2 5 was also 15 is given to be right.

(Refer Slide Time: 00:32)



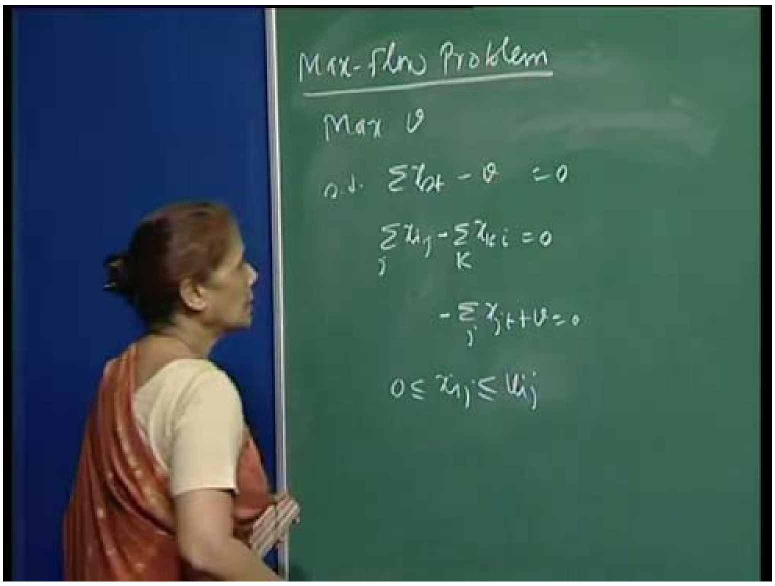
Suppose, I want to make u 2 5 go down to 10 units, text also I had discussed the situation when the capacity go up. But in this case now, for a basic arc, if the capacity goes down then, all that is happening is, see, you have node 2 here and node 5. So you were sending 15 units of flow here, from 2 to 5, because u 2 5 is 15, but we also had 15 units of flow from 2 to 5.

So, now if you want to reduce the capacity here to 10; that means, you have to find a path through which you can send 5 units, because you have to cut you you are not able to send more than 10 units here, but the b i's demanded that you have 15 units of flow here. So you have to now divert 15 units of flow from 2 to 5, so that is all. When the capacity went up, then you were trying to send, so you said that you will send extra amount from this node to this, and then, you will try to send the same amount - the increased amount from here to here.

So, now you just reverse, so that is it, so use that is what I said in the assignment when I was discussing the assignment, that you have the tools. So, the only thing you have to now consider is that, you need to send the reduced amount from 2 to 5 and you do the same steps that, for example, you will find a first path in the current tree from 2 to 5 and if you get if it has some extra capacity you will try to send some amount from along that path if some arc gets block there, then you will use the dual simplex step.

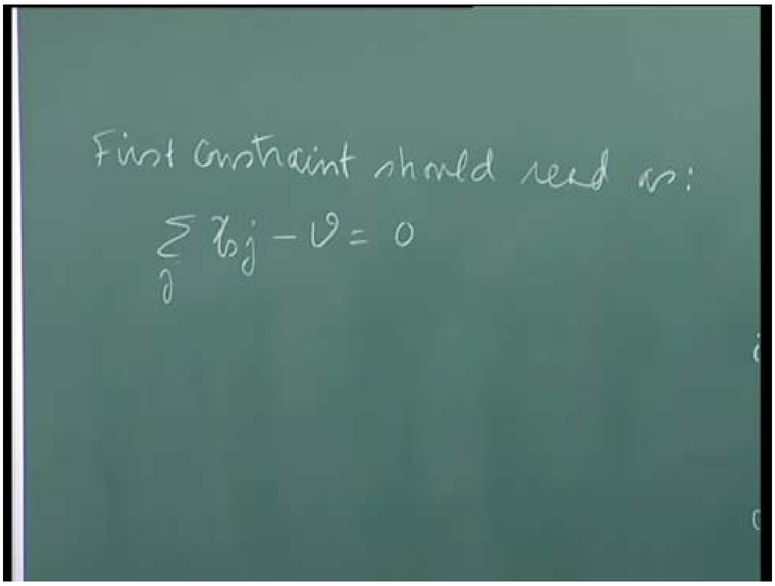
So, therefore, you can manage the thing, I was I just thought that maybe, I will give you they this hint and then you can work it out on yourself now. So, we were discussing the max-flow problem, and let me just revise quickly, we had define the problem, I had told you the given you the dual also, so we come back to now.

(Refer Slide Time: 02:51)



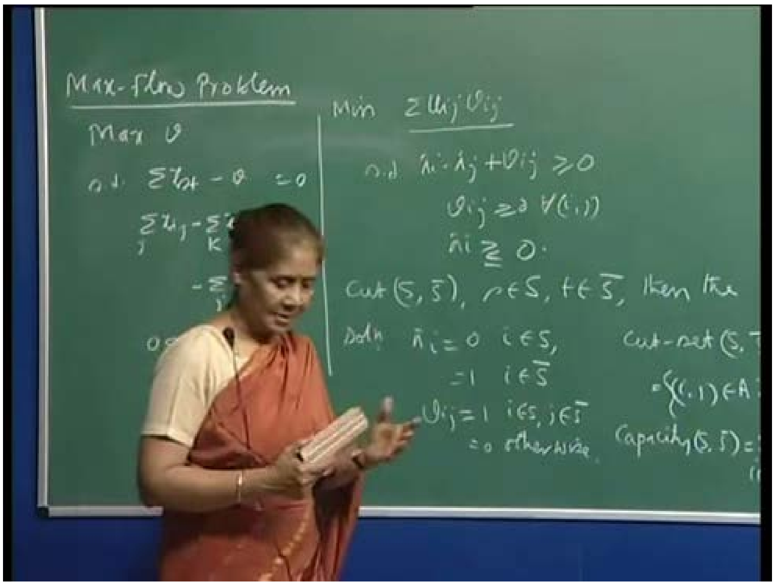
Max-flow problem, so our formulation was maximize v subject to sigma x s t minus v equal to 0, then summation x i j summation j minus summation x k i, because you are talking about node, i summation over k is 0 and then minus summation x j t summation over j is equal to minus v or ok - since we are writing it on this side, that is, make this that is what we are writing plus v is 0 - and then we had that x i 0 less than or equal to x i j less than or equal to u i j.

(Refer Slide Time: 03:47)



So, the first constraint should actually read as sigma x s j, because x is your starting node, so it should be x s j and summation over j. So, whichever nodes are connected to the starting node s for all those you have these variables x s j minus v should be equal to 0, this is the first constraint. Then, we wrote the dual to this problem or whether you want to call this the dual is upto you.

(Refer Slide Time: 04:11)



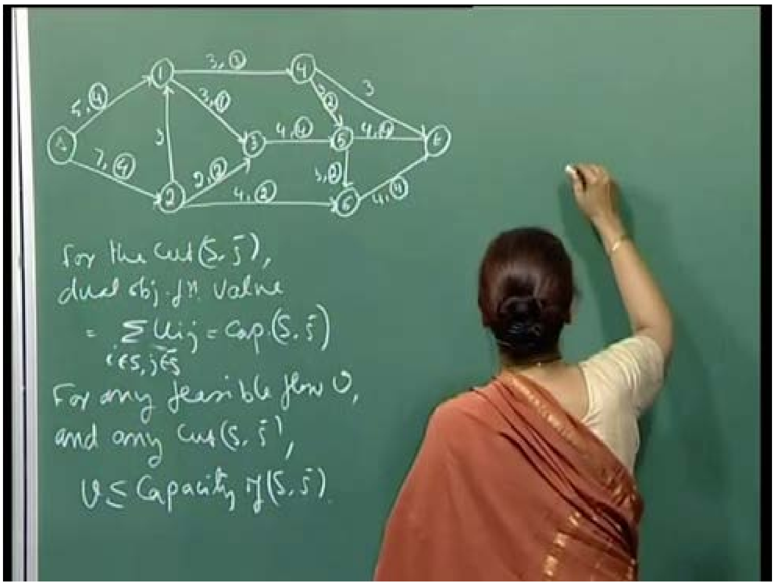
Anyway then, the corresponding dual problem is minimize, so you have only u i j's here of this type, so we said this will be minimization of u i j v i j. And then, for the x i j columns, you have plus 1 and minus 1, so subject to pi i minus pi j plus - the you I will not consider these as explicit constraints, only these ones.

So, you have v i j, and since your x i j's are non-negative, so you will minimization, this will be greater than or equal to, because x i j now, we are treating as greater than or equal to 0. So, these are my implicit constraints and therefore, the corresponding dual constraints would be greater than or equal to 0. And there is no sign restriction on the v's and so, so you will say that v i j's or greater than or equal to 0 for or all i j, and your pi i's are free - unrestricted.

So, this was the dual and then, I showed you that if you take a cut s s bar such that s belongs to capital S and t belongs to s bar then the solu[tion]- then the dual then the solution where pi i is 0 for i belonging to s, and is equal to 1 for i belonging to s bar, so which means that pi s will be 0 and pi t will be 1. And v i j is 1 for i belonging to s and j belonging s bar and 0 otherwise for all other cases.

So, we showed you that, we looked at it, we consider all possible cases of the dual constraints and i showed you this is a dual feasible solution. So, therefore, and So, that means, this is a cut and I define to you the cut-set s, s bar was collection of all i j belonging to such that i belongs to s, and j belongs to s bar. So, what are all arcs going from s to s bar, and see this is your cut-set right and then, we also defined the capacity of the cut. So, capacity of the cut s s bar is equal to u i j summation u i j i belonging to s and j belonging to s bar, so all the arcs which are going from s to s bar and they are adding up their capacities gives me the capacity of the cut. So, this is the dual feasible solution, and you see that for any cut the value of the objective function, so for the cut s s bar dual objective function value is actually equal to summation u i j which is the capacity of the cut, i belonging to s, j belonging to s bar which is the capacity of the cut s s bar.

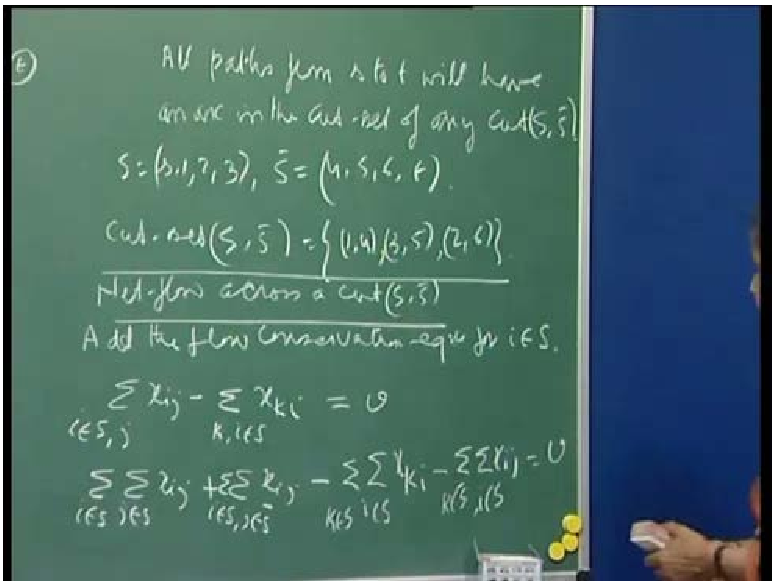
(Refer Slide Time: 07:38)



And by this thing, see it is a minimization problem, so I am calling it the dual, but by over tradition of whatever you call it, this was consider as the primal problem, but that does not matter. The important thing of the duality theorem is that the min objective function will be always greater than or equal to the max objective function.

So, here you see that if this is the flow, so for any feasible flow you have the value v. So, then, you immediately get that for any feasible flow v and any cut s s bar your v is less than or equal to capacity of s s bar. Now, as I said, we can look at cut in, so many ways, see when we because, you have partition the node set, and s is in one set, a one part of the partition and t is in the other partition, when you see all parts, so here I have drawn this graph and you can look at it, so what essentially another way to look at cut is that all paths from s to t will have an arc in the cut-set of any cut s s bar.

(Refer Slide Time: 09:31)



So, because it just divides the network into 2 disjoint sets therefore, any path, so for example, here you if we consider the cut, look at the cut s equal to 1, 2, 3 and s bar will be 4, sorry, this should include, so s is, I am taking this as the cut, so then in that case it is s 1, 2,3 and s bar will be 4, 6, 4 5 and 6 4 5 6 and t.

So, this is your s bar and your cut-set therefore, is cut-set s s bar is all arcs from s to s bar, and that is you see 1, 4 then 3, 5 and 2, 6, these are the all - the 2 6. And now you can just verify you for yourself that the cut-set is this, and any path from s to t will have one of these arcs in that path. So, therefore, since, the flow from s to t will be across all these paths from s to t, the total flow in the network cannot exceed the capacities of these arcs, because whatever flow you want to sent from s to t has to use arcs 1, 4, 3, 5 and 2, 6 right.

So, the total capacity when you add up of these three arcs, the flow cannot exceed this, so that is one way of looking at it, but since we have studied linear programming duality theory in detail therefore, one cannot, so immediately use the duality theory and say that the flow will always be less than or equal to the capacity of any cut and so by the duality theory whenever you have flow in the network and you have a cut, such that the flow is equal to the capacity of that cut then, this cut will be the minimum cut we will call i (( )) nomenclature and v will be the max flow.

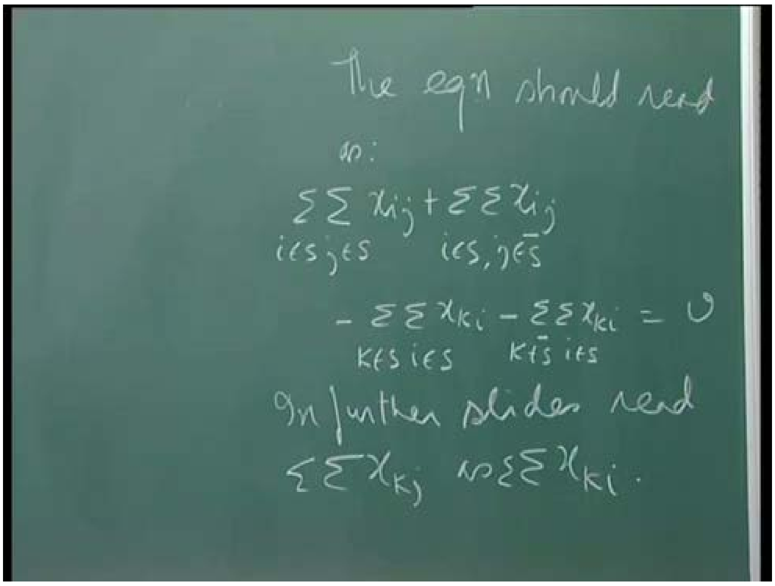
Because we have prove this from the duality theory that if for a feasible fair, the two objective function values are the same then, they must be optimal and the corresponding solutions are optimal, so we will continue developing on this, so this is your what we are trying to show here. And then, now, let me simply prove one simple result here, a net-flow across a cut across a cut s s bar.

So, now since for any feasible flow I want to show you what will be the expression for net-flow across a cut. And here what we do is in this case, just add up these flow conservation equations, so I will just write it out, add because this is for each i - for each node there is a constraint - so this for s and then this is for i, and then this is for the node t.

So, just add the flow conservation constraints, add the flow conservation equations for all i belonging to s only. So that means since, small s belongs to capital S, so this will be added and here, I will simply add these equations for corresponding to i in s. So, which means that I will get the equation, summation x i j i belonging to s and j of course, this minus summation over j and i belonging to - this is the - i belonging to s of x k i and this is equal to v. Because the first constraint has this number v here, which I write this way, so I am adding up this plus all other for all other i's for which belong to s, so part of them I will use, the remaining 1's I leave out, so this is the equation you get.

Now - I have written j, here this is k - now let us break up this summation because j and k are varying over the whole node set, so let me write it up as… so this will be summation x i j - or you want it, I will write the double summation, so this is i belonging to s and i belonging to j belonging to - because remember j is varying over the whole node set - so j belonging to s and plus, because this summation I am breaking up, so summation x i j double summation again i belonging to s and j belonging to s bar minus same thing here, summation double summation x k i, and here, this is summation k belonging to s and i belonging to s that is already there then, this one I am decomposing. So, minus summation double summation k belonging to s bar and i belonging to s is equal to - it is write x i j equal to v.

(Refer Slide Time: 16:25)

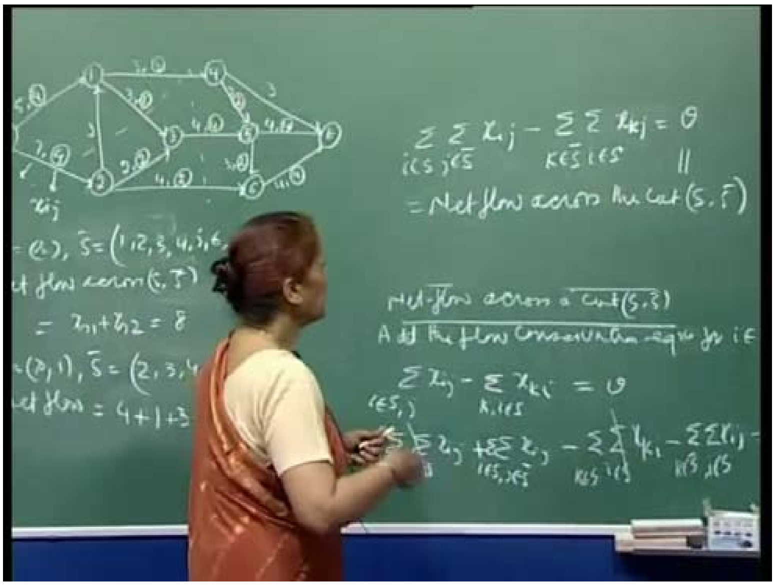


See, the equation when we are summing up over we are when we are summing up the flow conservation equations for all nodes in s then, the equation should actually read as, now this here I am breaking up the summation over v as s and s bar because s union s bar is v. So, therefore, the summation will read as summation over i belonging to s then, summation j belonging to s plus summation over i belonging to s and then, j belonging to s bar. So, that takes care of the first set of summation then minus similarly k belonging to s and i belonging to s then the second term is k belonging to s bar and i belonging to s. So, this all adds up to v, because we are adding up all, we are not adding up for the last node the last equation is omitted.

So, then I get this as equation and then of course, we further go on and analyze what I was computing the net-flow, and then, by mistake in further slides also I have been writing sigma double summation x k j, but actually it should continue to read as summation x k I, so the summation is over k and i. So, now, see here this and this cancels out, because this is summing up x i j and i and j both are in s and similarly, here also you are summing up all x i j's, where k is in s and i is in s, so these two cancel out.

And so you are left with the expression yeah let me just subtract this here we are looking at net-flow across a cut, so when this is this, this cancels out and you are left with summation i belonging to s, summation j belonging to s bar x i j minus summation double summation k belonging to s bar and i belonging to s of x i j is equal to v.

(Refer Slide Time: 18:10)



Why would i get k belonging to s bar and, yes, i belonging to s and ok, so this is what this gives you the flow from s to s bar and this will give you the flow, so the net-flow in the, so what we are saying is, see here maybe, I can draw the diagram here, I need that expression, see what I am saying is, this is s and this is s bar and your small s belongs here and your t belongs here. So, you have arcs going like this and then you have arcs going like this from here.

So, when we talk about the net-flow across a cut what we mean is that, see this much flow along the forward arcs is going from s to s bar, and some flow along the backward arcs; that means, arcs going from s bar to s there is some flow. So, the net-flow would be adding up these forward flows minus the backward flows this is what we mean. So, therefore, this is your net-flow across the cut, this is k belong, actually the arc here i j, where i is in s bar and where, sorry, where, sorry, k j, where k is in s bar and i is in s. So, you can immediately see from the figure that this is actually going from s bar to s these arcs.

So, you adding up these and subtracting, so therefore, this is the expression for, so this is equal to net-flow across the cut s s bar, and this is also equal to the flow which is in the network, because v is a total amount which is being sent from s, and since we are not violating these flow conservation equations, so whatever flow is sent from s will be reaching t.

So, the total flow in the network is v, but I have also shown you that the flow across cut will always be of, a flow across any cut will always be v. So, this will now be helpful to us, so maybe we can take up an example here just see. So, take any cut here and yeah so, now, let me come to this example. So, here the first number is the capacity, and this is your flow in that, so, I have shown you network here where with the arc capacities and flow that is there on the network.

So, if you look at say for example, look at the cut s simply this, and s bar will therefore, be 1, 2, 3, 4, 5, 6, t, and so what is the net-flow across this cut? Net-flow across s s bar, remember, we have to look at what are the forward arcs, so in this case they are no backward arcs, therefore, the net-flow is simply x s 1 plus x s 2 which is 8 units, this is the flow across the cut.

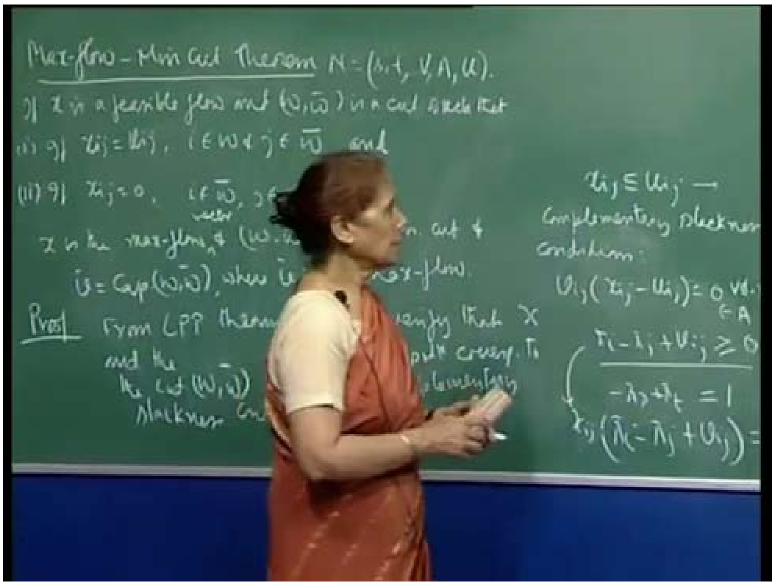
Now but if you look at, ok, let us choose some cut in which you have a backward arc. So, if I choose say for example this cut, now look at this cut, you have a backward arc here. So, therefore, here s will be small s s 1 and that is it, I am just using this 2, and then s bar will be 2, 3, 4, 5, 6 and t, this is a cut and let us see what is the net-flow across the cut, so here net-flow, so add up the forward arcs, so what are the forward arcs from s to s bar, from s to s bar, you have this, and you have 1 3 and 1 4; just see clearly, the forward arcs from this cut, so s s is connecting to 2 and then, 1 is connecting to 3 and 4.

So, the total flow is 4 plus 1, this thing and 1 4 is 3, so this is, is that ok, now I written 3 and plus right right right and this and then minus the flow across 2 1. Because this is the arc going from s bar to s, but that is 0, so there is no flow, so therefore, this is equal to 8 here which. So, you choose any cut whatever the flow in the network, if you are this thing is ok, these numbers are ok, and this is another way to check whether your numbers are alright or not, then the net-flow across any cut must always be equal to the flow in the network, so this is what the situation is.

And now let us look at, so I have already said what would be the min-cut max-flow theorem, but let us write it out and we can prove it in a nice way because you have this by now you I would say you have a strong back ground of the L p p theory. So, let us now write down the max-flow min-cut theorem, prove it and then, we will try to sum with an algorithm, yeah and then, yes, let us do that. And I have already gave you the interpretation of the primal dual algorithm, I showed you that. A primal dual algorithm will actually reduce to finding an augmenting path from s to t, and then trying to augment the flow.

So, this is what we are going to now formalize, after I prove the min-cost flow theorem, I will formalize the algorithm of building up the max-flow by a selecting by finding augmenting path that each iteration, so let us work out this details.

(Refer Slide Time: 24:59)



Max-flow min-cut theorem, this is that if, so we have a network n where your source and sink nodes are specified, v is the node set, a is the arc set, and you denotes the capacities on the arcs, u is the vector of capacities. And we say that if x is a feasible flow and w w bar is a cut such that.

So, I am given a feasible flow and I am saying that I have a cut w w bar such that if x i j is equal to u i j, then i belongs to w and j belongs to w bar. And second condition if x i j is 0 then, i belongs to w bar and j belongs to w, so that means, when it is a backward arc in a sense with respect to the cut w w bar then if this flow should be 0, then x is the optimal flow or max-flow I should say, is the max-flow and w w bar is the min-cut and x is the x is the, here again we understand should be clear x is the max-flow vector I should say, max-flow vector, because the components here denote the flow on the arcs. So, the maximum vector w is the min-cut and v bar is capacity of w w bar where v bar is the max-flow idea. I just want to state the theorem because may be it can be written in a better way, but anyway. So, then the value of the max-flow will be equal to their capacity of the cut and this will be the minimum cut, because remember the dual problem we were defining as finding the minimum of sigma u i j v i j.

So, this will be the minimum cut and this will be max-flow, the proof is simple, I have now rubbed out this thing. But remember, in your formulation of the max-flow problem, the constraints were all equality except, see, you had x i j less than or equal to u i j, these are the only inequality constraints. So, if you want to look at complementary slackness conditions when this required that your v i j into x i j minus u i j should be 0 for all i j belonging to a.

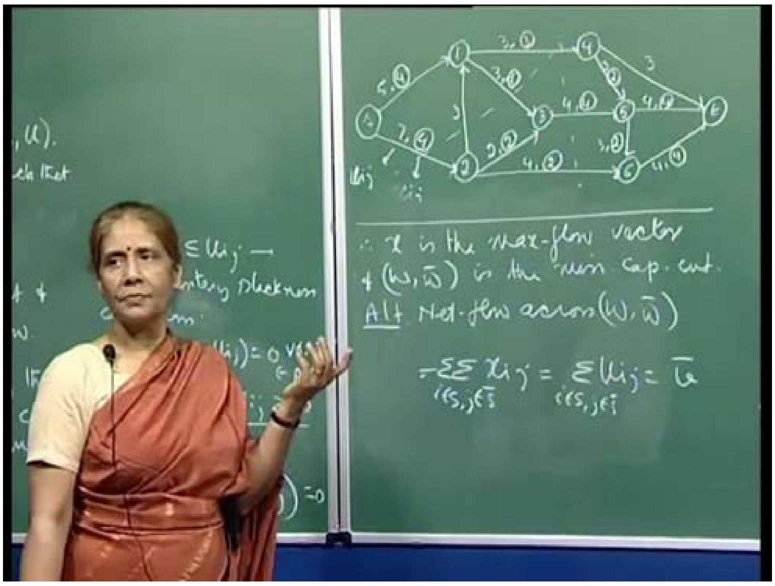
So, this will be the corresponding to the primal constraints, this will be your complementary slackness conditions, because what means that? If x i j is u i j and your v i j is 0, or v i j can be anything sorry, but when x i j is 0 then v i j must be 0, because when x i j is 0 u i j this is anyway 0, so it does not matter what v i j is, but when x i j is not equal to u i j; that means, less than u i j which means it is 0 then v i j must be then v i j must be 0. And this is what we are defining here, we are saying that remember, for the dual solution when you have a cut w w bar we are defining the dual feasible solution v i j is 1 when i is in w and j is in w bar and when it is the other way then, v i j is taken as 0, so that was required for the dual feasibility.

So, therefore, here actually one does not need the proof as search in the sense that all we need to say is that from L p p theory we can verify that x and the corresponding dual and the and the dual feasible and that and the dual feasible solution corresponding to the cut w w bar satisfy the complementary slackness conditions - Maybe, I should also, slackness conditions yeah you - You have to do the work, because remember for the dual constraint I had pi i minus pi j plus v i j greater than or equal to, yes, when I wrote the dual constraint I forgot about the, I did not write them, I will do that immediately, I will. So, make that correction, this is greater than or equal to 0 and for the variable v, I did not write the dual constraint, which is minus pi s plus pi t and this is because v is unrestricted comes out to be one.

So, this is anyway always satisfied, because we choose pi s to be 0 and pi t to be 1, so this is always be satisfied, so again no problem, but for. And this one would be pi i minus pi j plus v i j, so here the complement is slackness condition would be corresponding pi i minus pi j plus v i j into x i j this should be 0, for all i j, and that you can see from here that when you have i is here. In fact, you would need to take all possible cases as I discuss for dual feasibility, when I was trying to prove that this dual feasible solution. Then, you just see that here I will take up one case that i is here, j is here, then your pi i is 0, pi j is 1 and this is also 1, so therefore, this is 0, so then x i j can be positive which is happening, because then x i j is u i j according to here, and now you can also discuss the other possible cases.

So, what I want to say is that if an x and the corresponding and a cut w w bar satisfy these 2 conditions then the complementary slackness conditions are satisfied and therefore, we immediately get that, the two solutions will be optimal; that means, x is the max-flow and w w bar will correspond to the min-cut, now you want to show the value of the, what will be the value of the cut, so therefore, here and yeah you want to look at the…

(Refer Slide Time: 33:40)



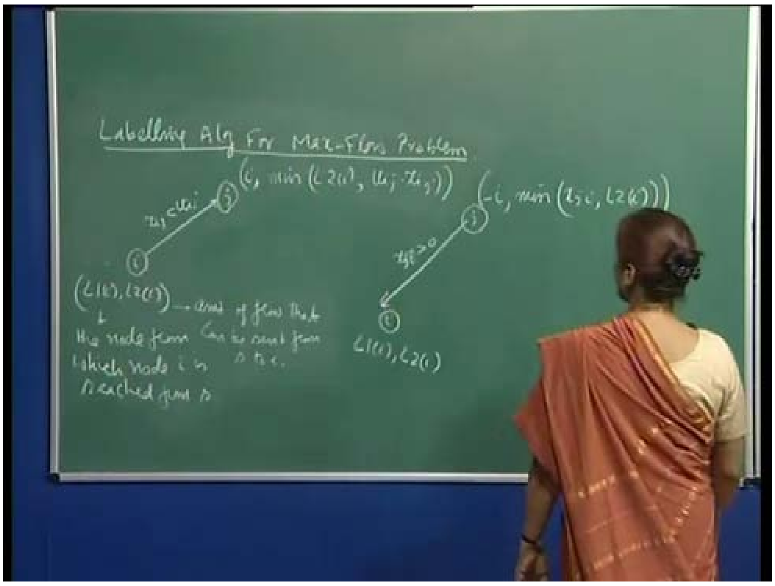
So, therefore, the complementary slackness conditions therefore, x is the max-flow vector and w w bar is the minimum capacity cut, this comes from our complementary slackness conditions, but you can otherwise also c or otherwise alternatively you can also check see we said that the net-flow across a cut is given by this.

So, alternatively net-flow across w w bar is equal to what by this definition, because these conditions are being satisfied, so you see for i in w j in w bar x i j is u i j and this is 0, when it is the other way. So, here when you write down this, you see this flow is 0, because for k in s bar and i in s the x k j is are all zeros. So, therefore, this is 0 and so you get that net-flow across w w bar is simply summation x i j i belonging to s, j belonging to s bar which is equal to summation u i j i belonging to s, j belonging to s bar and that is equal to your flow by that equation.

So, alternatively also you can proof it, because you will find that some text books may not use linear programming theory to prove all these results which is fine, but I feel that you have a good background of linear programming theory by now and therefore, we can also see that you can easily prove this result. So, this is equal to v bar and so once the because we said that the a flow across any cut the; the flow in a network will always be remember, the v duality theorem, I already showed you that the flow across, flow in network will always be less than or equal to the capacity of any cut.

Now, once the flow is equal to the capacity of a cut then, they both must be optimal right. So, we have shown that the maximum flow will always be equal to the minimum capacity of the cut which has minimum capacity. So, this almost gives you the theory behind the and, so this will be our stopping criteria also, when we want to show that we have a flow; that means, now when we develop an algorithm our stopping criteria will be that the moment we have an flow which satisfies these conditions and we can determine the cut w w bar; that means, if given a flow, I can find out for you w w bar, so that these conditions are met, then I will stop, because I know that the flow must be maximum. So, now, the idea is to develop an algorithm and this will be very simple algorithm i will give you and because the course does not really have it in it scope to give you, you know discuss have a lengthy because otherwise, we might end up having 7 to 8 lectures on discussing the a real state of arc max-flow algorithm.

(Refer Slide Time: 37:33)



So, let me just give you an idea here, so I will be giving with the labeling algorithm for max-flow problem. So, the idea here is, see let us look at the how, so if you have a node i and that than arc like this j, so we will give the label here two labels l1 and whatever the this thing L 1 i and then L 2 i, so these will be the labels associated with any node L 1 i will indicate…

So, this will be the node from which node i is reached, this is the node from which node i is reached from s. Remember, we want to, as I told you the basic idea is to build up paths which have positive capacity, so that i can flow certain amount on that arc on that the path. So, this will indicate the node from which, somewhere here, from which I am reaching i and then that node is reachable from s.

So, therefore, essentially we start from s and I am finding the path, Now if x i j is less than u i j, then I will label this as i and this will be minimum of - I will explain - L 2 i and u i j minus x i j. So, this will be minimum of this and so this will be the label. So, as I said because, now you have an arc from i to j which has some capacity left on it, because the flow currently is less than u i j, I can increase the flow on this arc. So, therefore, I label the first label as i, same thing because j is now reachable from s, I can come up to this point, and then the second label tells you the amount of flow that can be flown up to this point. So, L 2 i was the. So, i should have said here L 2 i is the amount of flow but can be sent from s to i.

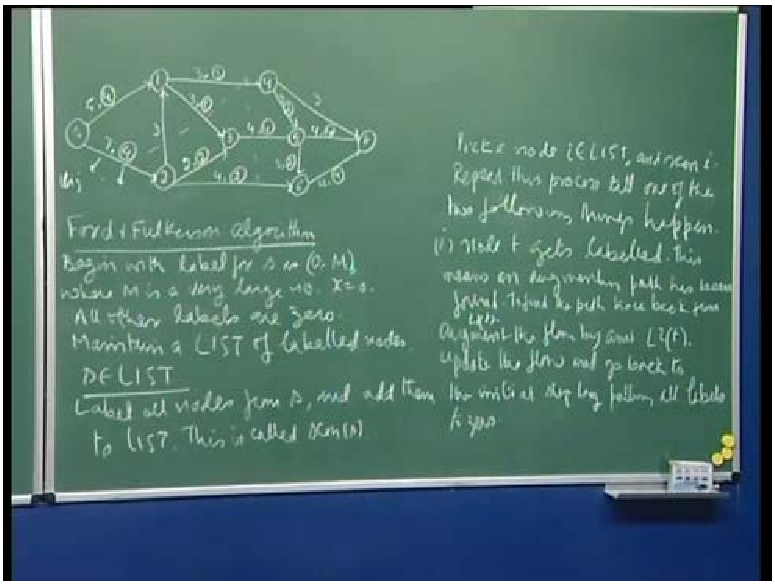
So, let us just (( )) I was just telling you how to label a node, from a node which has already been label, so the 2 labels. The first label indicates the node from which i is reachable from s. So, through that node you come to i and then l 2 i is the amount of flow that can be send from s to i.

So, now when I want to label j, in case x i j is less than u i j, the first label will be node I, because j is reachable from i and so the path that was coming from s to i can now come up to j. And then the amount here which can be sent from s to j, obviously, has to be the minimum of the amount that was already been, can be sent to i and then, the amount which is can be sent on this arc, so the minimum of the two, remember, I defined for you the capacity of a path as the minimum capacity of an arc on the path.

So, therefore, here this number has to be minimum of the two and then, this will be the label. In case, your arc is like this, so i has been labelled and this is j, so here you will require that x i j should be greater than 0, because when you want to label j from i that means you will be reducing the flow on arc j i essentially. So, if there is a positive flow from j to i, when I traverse it in the opposite direction from i to j, I will be reducing the flow. So, here, if this thing is L 1 i, whatever it is and this is L 2 i a label, so here the label will become minus i.

See here, I did not write plus i it is understood, but minus i means that you are traversing the arc in the opposite direction; that means, the arc is actually from j to i, so I write minus i and then this will be minimum of, again, the same thing I can reach this much flow up to this point from s. So, now here to be minimum of x j i or i should have said here that j i, j i is positive, because flow is actually from j to i. So, minimum of x j i and no l to i, so this is your label, so once you have this method of labeling I know already labeled. So, then we start looking for, so here, let me give you the algorithm, so this is ford and fulkerson’s algorithm actually.

(Refer Slide Time: 43:15)



Ford and fulkerson algorithm, so begin with label for s as 0 m, where m is a very large number. See, we have no idea how much flow will be able to send from s, so we will just leave it as 0 m, so this is thing. Then, label all nodes, and initially where m is a very large number and 0 flow in the network, so with that we have to say, and we can say that x is 0 and all other labels are 0.

Then, you start, so you maintain a list of labelled nodes, so this is all your initialization start with all 0 labels except a node s which has the label of 0 and m, then currently the flow in the network is all 0, you have a list of labelled nodes, so in the beginning right now, so s belongs to this to start out. Then, idea that you and now, you label all nodes from s by our method and add them to list and this is called scan s.

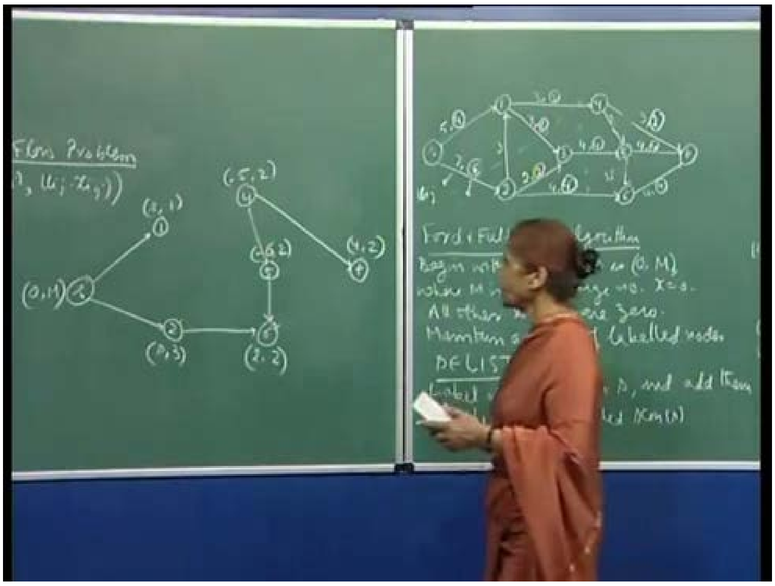
Scan scanning node means that it is already labeled, it is in the list, you take it out from the list and then you label all possible nodes from that node, and add them to list, so this is what you called, this is called scan s. Then, the second step will be, now pick a node i belonging to list and scan i, so repeat the process, when I scan i again whatever nodes i can labeled from i, I will add them to list.

So, repeat this process till one of the two following things happen. See, remember, the idea is to find an augmenting path, so if I label 2 follo[wing]- for 1 of the 2 following things happen, one is node t gets labeled, node t gets labeled right; this means, an augmenting path has been found and so augment the flow by amount L to t, remember, the second number indicates the amount flow that can be sent from s to t. So, augment the flow by among this thing, update the flow, augment the flow, but, so an augmenting path has been found and how do you find this. So, this is to find the path, trace back from L1 t, I will just explain to in a minute; see that means, because this node tells you this label tells you how you reach t, and when you go to that load that will tell you how you reach there.

And so, as you go backwards, so find the path since trace back from L 1 t and you will find the path up to c. So, you will find the augmenting path, then on the augmenting path you will augment the flow by amount L 2 t. And then, update the flow that means, you will change these numbers and update the flow, and go back to the initial step by putting all labels to 0.

So, all that work you did in the first iteration is not going to be used at all, you will need to start labeling the nodes again and again find a augmenting paths. So, let us look at for example, here in this case, let us look at this example, you have this and let us try to start labeling from node s. So, I will use it here because yeah, so of course, what is the label. for s, so s is 0 m I am starting with then if you look at node 1, this arc, this is not saturated is a flow of 1 unit of available.

(Refer Slide Time: 49:51)



So, here the label would be s and then the difference is because this is m, so the minimum of m and 1 will be obviously 1, because m I am saying is a very large number, so this is the label for node 1. Similarly for node 2 the label again it is reachable, s 2 is not saturated there is a capacity of 3 units left on this arc and so the label would be s and 3 again because m is a large number. Then from 2, 1 is already label, so you cannot label, so let us look at 2 3 2 6, so 2 3 is what? 2 3 is this 1 here, the capacity ok, I cannot label 3 from 2 because this is already saturated remember, for forward arcs if the arc should not be saturated, so I am not able to label this here.

Then, 2 6 you can label because it is not saturated, and this left over capacity is 2, so therefore, this will be 2 and 2, minimum of 3 and 2 will be 2, so this is what. So, this is what see your building up the path and it tells you that starting from s you can come to 2, then from 2 you can go to 6 and the flow that can reach up to 6 from s is 2 units right. Now, and then if you like you could have see, that is what I am saying here in the algorithm nothing is spelled out is to how you will operate the list.

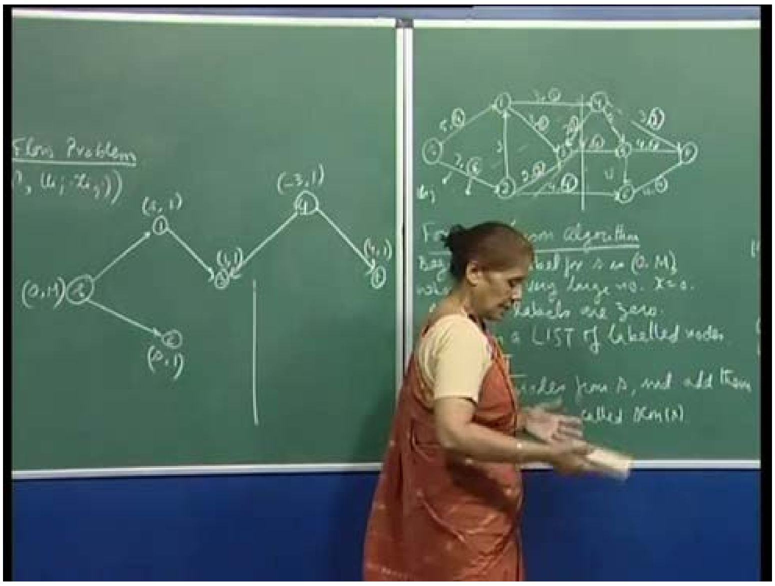
See, all it says is, pick a node; pick a node i from list, nothing is said about it and therefore, I could have for example, chosen from here to label t, and the label t I cannot do it right, so I will means, I will stop here and from 6 now if you can you can label 5 right, so let us do it, from 6 you can label 5 which would be, sorry, the arc is this way and flow is 2 here, this is the arc I will label 5 here, but what will be the labels see remember its backward, so it will be minus 2 and 2, because I can reduce the flow by 2 units 2 units is here hence the minimum of this and this number is 2, so I can do it this way right.

Now, from 5 I cannot label t and I can label 4, so which is again a backward arc, so write it this way 4 and here again the label would be minus 5 and the flow is 2 again, so this will be 2, and then from 4 you could label t and what would be the, because this is there is no flow on 4 t, so therefore, the minimum of 2 and three is 2, so this tells you that you can have 4 and 2. So, therefore, the augmenting path has been found which is s 2 6 5 4 t and since that you can augment the flow by 2 units and as I said once you have label t you can trace back the path because this tells you came to node t by from 4; 4 tells you that you came by 5 by a back backward arc.

So, you come here 5, why did I write minus 2, this is minus 6, so this is minus 6, because I came from 5, I have I came to 5 from 6 by a backward arc 6 of this 1 has a label 2 which says that you came to 6 by 2 and 2 i of course, came from s. So, this is how I trace the path this number tells me the capacity of the path, so I can augment the flow by 2 units and so therefore, these numbers will go up, the here this number will go up to 6 and this will go up to 4, and then this number becomes 0, there is no point writing circle a because we are not talking of a basic feasible solution or anything for this and then here, this is 2, this is how I update my flow, so all labels are erased.

Now, I will start again and let see what will be my labeling scheme, so here again s 1 is the same thing because that flow did not change then 2 can also get labeled again, so this will be s and 1 right as a unit flow then 2 3, I cannot label sorry 2 3 yeah, now, I can label - why I did not label, 1 is already labeled fine. Now, from 2 I cannot label 3 or 6 right, so I will go look at node 1 here.

(Refer Slide Time: 54:59)



Now, node one this is saturated the only arc available is 1 3, so we will do this, and here the first label is 1 and the second label is one, because this flow is 1 unit and also what you can reach to one from s is 1 unit, so this is 1 1 right. And from 3 we could label while that is it, you cannot go for any further, because from 1 to 3, I could just increase the flow by 1 unit this was 4, so 1 unit is there, so that is right, so I cannot do anything here and that is it, because there is no backward arc for which I can reduce the flow, see this is already labeled no who is labeled. So, I could go from here to here, but am not doing it, and then in any case this is also saturated right, so this is what to have, so that you have to 8 and 2 - 10 units of flow, you have 10 units of flow fine.

So, now, what we have you, will show you is that once the algorithm stops; that means, the go back to the, so I did not tell you how to fine, so now we will come to; that means, you go on augmenting, so this was the an example to show you how you will find an augmenting path and then, augment the flow, so then what will be the termination criteria, the algorithm stops - I will just write - the algorithm stops when node t cannot be labeled, right and this is what I will prove to you. So, let us quickly do it, see here we could label 1 2 and 3. So, therefore, what I am now going to show you is that my cut, this will define my cut ok, so the nodes which get label, so when the ford and fulkerson's algorithm stops and t is not labeled, so therefore, all nodes which have been labeled, I will call them as a part of s, and the other nodes will be part of s bar t; obviously, will be in s bar s is always in s.

So, now you see that if you look at this cut, see the capacity of the cut is 4 4 and ok, this is 3 oh so that means, there is something missing so; that means, we will because this is a good check; that means, the algorithm should not stop here right now, I can label something more, because the capacity of the cuts right now, a currently the flow is 10 units, but that means I should be able to… so I did not somebody correct, so here some we will look at the problem again, because my this thing is this the cut is this not, this one here.

So, the cut is this, all the three arcs are saturated 4 4 and 3, so the total flow should be 11 units may be this, this one I did not write the number correctly where, so will revisit the problem because here this one is showing that the capacity of the cut is the 11 if this is the cut then I should be able to show that is 11 units of flow in the network; that means, we should be able to increase the flow by one more unit ok.

So, we will see, we will correct the calculations needs to be check, this arc was missing in the network that is why you see whenever you start with the problem always make sure that the flow conservation equations are satisfied and so therefore, arc 4 3 is there and then see the 3 units of flow was coming from here only two was going out here, so one was going here this was needed right.

So, once this is there, you see then I can label 4 from 3, so this stopping algorithm criteria were given, and so from three you can label 4 and this will be minus 3 and one right because you have flow of 4 3, so this will be this and so again you can find an augmenting path which would be 3. So, this will be the of course, right now the label would be 4 and 1 ok.

So, once again you have to label t and the augmenting path is, you coming here from 4; 4 to this is this, so therefore, this is the backward minus 3, then you are coming to from here 1 and from 1 you are coming from s, so this is another augmenting path and so your total flow will go up to now 11 units and therefore.

See, after this you will see that we will continue with the now you try to find out once you augment the flow that the cut will be, that you once when you cannot any more label the nodes when the label nodes you will form as s and the others as s bar, and they will form the cut for which the capacity will be equal to the flow in the network.